LEVERAGING INFORMATICS FOR ACCELERATING THE DISCOVERY OF MATERIALS

by

Alberto Hernandez

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

April 2021

Abstract

The application of materials informatics for the rational design of materials has been inspired by the increasing number of examples of success of machine learning in many fields, and it has been facilitated by the greater access to computational resources, the advances in algorithms and the growing open-source code community. This thesis presents two ways in which we have advanced the field of computational materials science through materials informatics. A promising application of materials informatics to materials science is the development of machine-learned interatomic potentials models that are orders of magnitude faster than *ab initio* methods such as density functional theory and can be nearly as accurate. However, these models are typically orders of magnitude slower than physics-derived models such as the embedded atom method (EAM), and they usually do not generalize well. We present a supervised machine learning approach for developing interatomic potential models to simulate atomic systems at large time and length scales from *ab initio* data. The models developed with our symbolic regression algorithm are computationally fast, simple (and interpretable), accurate, and transferrable. A reason for the success of our algorithm is that it learns models using a physics-informed hypothesis space. Another important component of our algorithm is the minimization of a multi-objective cost function to search simple, accurate and fast interatomic potential models. We first demonstrate our approach for elemental Cu, and then show how the models discovered for Cu transfer well to other fcc transition metals close to Cu on the periodic table. Then, we demonstrate how our algorithm can be used to discover new functional forms for the fcc transition metals close to Cu on the periodic table, benefiting from the information encoded in known models as a seed to the search. The machine learning

interatomic potential models developed with our approach are 2-3 orders of magnitude faster than other machine learned potentials, they are on average one order of magnitude simpler than EAM-type models, and their transferability is at least as good as that of other EAM-type models. In addition, their simplicity opens the door for studying their functional forms to possibly gain insights into the atomic systems. This thesis also addresses the need for a database of atomically precise nanoclusters at the density functional theory level of accuracy. Our approach used a genetic algorithm to identify low-energy clusters, and to our knowledge, it constitutes the largest database of atomically precise nanoclusters at the level of accuracy of density functional theory. This database can inform studies that aim to design clusters for a variety of applications, it can be used to train machine learning models, or it can be used as a benchmark for other studies.

Thesis Advisor: Prof. Tim Mueller

Thesis Committee:

- Prof. Michael Falk
- Prof. Paulette Clancy
- Prof. Jeffrey Gray
- Prof. Kit Bowen
- Prof. Tim Mueller

Acknowledgements

I am profoundly grateful to Professor Tim Mueller, my PhD advisor, for his guidance during the PhD program. His unparalleled support and encouragement helped me make progress towards my PhD. With Professor Mueller I have learned how to address scientific research problems and communicate findings effectively. The lessons that I have learned with Professor Mueller during my PhD have helped me become a better scientist and researcher. I also want to thank Professor Michael Falk, Professor Paulette Clancy, Professor Jeffrey Gray, and Professor Kit Bowen for their time and feedback as members of my thesis committee.

I want to thank Dr. Sukriti Manna and Dr. Fenglin Yuan, two of my co-workers and mentors at the Mueller Research Group. Their passion as Postdoctoral Associates inspired my own work as a PhD student. I am grateful for the help and commendable work ethic of my research colleague Adarsh Balasubramanian. I would like to thank other members of the Mueller Research Group for their comradery, especially Dr. Liang Cao, Dr. Tanmoy Chakraborty, Dr. Pandu Wisesa, Dr. Shanping Liu, Dr. Chenyang Li, Chuhong Wang, Yunzhe (Phil) Wang, Peter Lile, Thomas Nilson, Hao Gao, Wan Wan, Sam Norwood, Simon A. M. Mason, and Koutaro Aoyagi. I would also like to thank my collaborators, Adam Peters and Professor Jim Spicer for their support. I acknowledge the help received from Jeanine Majewski, Ellen Libao, and Tanea Melvin, members of the Administrative Staff in the Department of Materials Science and Engineering. Also, thanks to my friends, who were supportive during this journey.

I acknowledge financial support from the Office of Naval Research, grant number N000141512665. I thank the services provide by the high-performance computing

resources from the Maryland Advanced Research Computing Center (MARCC) and the Cray XC40/50 (Onyx) supercomputer from the Army Engineer Research and Development Center (ERDC).

Finally, I would like to thank my wife, son, parents, siblings, and extended family for their love and support.

Dedication

Dedicated to my family.

Table of Contents

Abstract ii

Acknowledgements iv

Dedication vi

Table of Contents vii

List of Tables x

List of Figures xii

1 Introduction and Overview 1

2 Fast, Accurate, and Transferrable Many-Body Interatomic Potentials by Symbolic regression 4

2.1 Background 4

2.1.1 The Potential Energy Surface 4

2.1.2 Interatomic potential models 6

2.1.3 Symbolic regression via genetic programming 15

2.2 Introduction 16

2.3 Methods 19

2.3.1 The hypothesis space of the supervised learning problem 19

2.3.2 Description of the artificial intelligence algorithm 22

2.3.3 Details about the target data 25

2.3.4 Details about the interatomic potential models from the literature 27

2.3.5 Enabling GP1, GP2, and GP3 in LAMMPS 30

2.4 Results 30

2.4.1 Validating the machine learning algorithm 30

2.4.2 Testing the symbolic regression algorithm: discovering new models for copper from *ab initio* data 33

2.4.3 Assessing the transferability of the interatomic potential models 41

2.4.4 Analysis of the functional form of GP3 59

2.4.5 Benchmarks of computational cost 61

2.5 Discussion and conclusion 62

2.6 Data availability 65
3 Generalizability of the Functional Forms of Interatomic Potentials Discovered using POET 66
3.1 Introduction 66
3.2 Methods 68
3.2.1 Developing the interatomic potential models 68
3.2.2 Density functional theory data generation 75
3.2.3 Computing properties with interatomic potential model 77
3.3 Results and discussion 78
3.3.1 Assessing the transferability of functional forms developed with POET for Cu to other elemental systems. 78
3.3.2 New functional forms identified using POET. 94
3.3.3 Assessing the tradeoff between accuracy and complexity: validating against literature EAM-type models 100
3.4 Conclusion 105
4 Developing a database of atomically precise nanoclusters 106
4.1 Background and summary 106
4.2 Methods 109
4.2.1 Identifying low energy clusters using a Genetic Algorithm 109
4.2.2 Correlations between elements in the Quantum Cluster Database 113
4.2.3 DFT calculations 116
4.2.4 Workflow 117
4.3 Data records 121
4.3.1 File format 122
4.3.2 Properties 123
4.4 Technical validation 124
4.5 Usage notes 125
4.6 Code availability 128
4.7 Acknowledgements 128
4.8 Author contributions 128
5 Conclusions and Outlook 130
5.1 Interatomic potential models by symbolic regression 130

5.2 Developing a database of atomically precise nanoclusters 132
6 References 134
Vita 155
Appendix A 156
Appendix B 173

List of Tables

Table 1. Acronyms used for the interatomic potential models. 29

Table 2. The 3-dimensional convex hull of models found by the machine learning algorithm. 35

Table 3. Errors on different properties for models on the 3-dimensional convex hull. The models are listed the order they appear in the table. C_{ij} are elastic constants, a_0 is the lattice parameter, $\Delta E_{bcc\text{-}fcc}$ is the energy difference between bcc and fcc phases, E_v is the fcc bulk vacancy formation energy, $E_{v,\ unrelaxed,\ 2\times2\times2}$ is the unrelaxed vacancy formation energy computed on a 2×2×2 supercell, E_m is the migration energy for fcc bulk vacancy diffusion, E_a is the activation energy for fcc bulk vacancy diffusion, $E_{dumbbell}$ is the dumbbell <100> formation energy, ν is the phonon frequency, and γ_{ISF} and γ_{USF} are the intrinsic and unstable stacking fault energies, respectively. $\bar{\gamma}$ is the average surface energy weighted according to the Wulff construction and γ_{abs} is the mean absolute surface energy over 13 surfaces. 37

Table 4. EAM-type interatomic potentials for Cu near the Pareto frontier of maximum absolute percent error on elastic constants and complexity. The Pareto frontier can be defined as follows: no model is both less complex and has less error than a model in the Pareto frontier. 39

Table 5. Errors on elastic constants and lattice parameters of EAM-type interatomic potentials for Cu 44

Table 6. Error of the values predicted by EAM-type interatomic potentials for copper relative to the respective reference. The models displayed in this table are near the Pareto frontiers in Figure 6, values of other potentials are in Tables S2 to S7. C_{ij} are elastic constants, a_0 is the lattice parameter, ΔE (bcc-fcc) is the energy difference between bcc and fcc phases, E_v is the fcc bulk vacancy formation energy, $E_{v\ (unrelaxed,\ 2\times2\times2)}$ is the unrelaxed vacancy formation energy computed on a 2×2×2 supercell, E_m is the migration energy for fcc bulk vacancy diffusion, E_a is the activation energy for fcc bulk vacancy diffusion, $E_{dumbbell}$ is the dumbbell <100> formation energy, ν is the phonon frequency, and γ_{ISF} and γ_{USF} are the intrinsic and unstable stacking fault energies, respectively. 47

Table 7. Errors on difference between energies of FCC, BCC and HCP phases of EAM-type interatomic potentials for Cu..... 48
Table 8. Comparison between genetic programming potentials and a neural network potential.[151]..... 49
Table 9. Errors on bulk vacancy formation energy, migration energy, activation energy and dumbbell <100> formation energy of EAM-type interatomic potentials for Cu..... 51
Table 10. Errors on phonon frequencies of EAM-type interatomic potentials for Cu 53
Table 11. Prediction errors for the intrinsic stacking fault (γ_{ISF}) energy and the unstable stacking fault (γ_{USF}) energy..... 54
Table 12. Prediction errors for surface energies of EAM-type interatomic potentials for Cu 55
Table 13. The 3-dimensional convex hull of models found by seeding with GP1 and GP2 and including the 13 low-index surfaces in the training data 58
Table 14. Acronyms of the interatomic potential models discussed in this chapter..... 69
Table 15. Initial parameters of SC models [138]..... 70
Table 16. Pseudopotentials used in VASP..... 76
Table 17. Temperatures of the DFT molecular dynamics simulations used for generating the training and validation data..... 77
Table 18. Cutoff distances used for the interatomic potential models for each element.. 78
Table 19. Parameters of GP1 models..... 82
Table 20. Parameters of GP2 models..... 83
Table 21. Parameters of GP3 models..... 83
Table 22. POET GPn models (see appendix for full numerical precision). 95
Table 23. Literature references of the experimental vacancy formation energies used for determining the maximum and minimum experimental vacancy formation energy for each element..... 100
Table 24. Keys, types of data, and description of the QCD data in the JSON file and .csv format..... 122

List of Figures

Figure 1. Crossover operation in genetic programming. (a) Trees selected for crossover. (b) Offspring after the crossover operation. 16

Figure 2. Tree graphs of a) Lennard-Jones potential parametrized for argon, equation (2.14) , b) Sutton-Chen EAM potential parametrized for copper, equation (2.16), c) GP1 and d) GP2 21

Figure 3. Example of a crossover operation 22

Figure 4. Parity plots of training (orange) and validation (blue) energies, components of force and components of the virial stress tensor for the interatomic potential GP1 (a) and GP2 (b). The black dashed line is the identity. The mean absolute error (MAE) is presented above each sub-figure for validation and training data respectively. 42

Figure 5. Radial distribution functions of liquid copper at 1400K 43

Figure 6. Pareto frontiers of EAM-type interatomic potentials for copper. No model has less error and is less complex than a model in the Pareto frontier. The orange dashed line was the Pareto frontier before the development of GP1 and GP2, and the blue dashed line is the new Pareto frontier. The percent error for each model was evaluated against the model's own target values, described in the Methods section of this chapter. Complexity was measured by the number of nodes in the tree representation of the model. Because the smoothing function for some models is unknown, to construct this plot each smoothing function was counted as 2 nodes, representing the smoothing function and a multiplication operation. Sources: SC[138], ABCHM[140], Cu1[140], EAM1[123], EAM2[123], Cu2[142], Cuu6[143], Cuu3[144] and CuNi[136]. The interatomic potentials were found in the Interatomic Potentials Repository.[37] GP1 and GP2 were developed as part of this thesis. 45

Figure 7. Average (instead of maximum) absolute error on elastic constants. The Pareto frontier does not change from the one calculated using maximum error. 46

Figure 8. Calculated phonon dispersion curves for DFT, GP1, GP2, and GP3. 53

Figure 9. Surface energies of elemental copper as computed using DFT, and the interatomic potentials GP1, GP2, and GP3. 56

Figure 10. Tree representation of GP3. 60

Figure 11. Different components of the potential model GP3. A repulsive interaction (right axis) is shown in blue, the attractive interaction (right axis) in green, and the smoothing function (left axis) in brown..... 61
Figure 12. Computational cost of potential models in LAMMPS. SC (5) uses a cutoff distance of 5 Å, SC uses a cutoff distance of 10 Å. The cost is similar for SC (5), EAM1, GP1, GP2 and GP3. 65
Figure 13. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Cu. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model. 72
Figure 14. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Ag. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model. 72
Figure 15. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Au. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model. 73
Figure 16. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Ni. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model. 73
Figure 17. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Pd. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model. 74
Figure 18. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Pt. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model..... 74

Figure 19. Convex hull of “selection value”, see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Rh. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model. 75

Figure 20. Convex hull of “selection value”, see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Ir. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model..... 75

Figure 21. Error on the validation set of energies, forces and stresses in logarithmic scale with base 10. This error metric is the same as the fitness. The fitness is the weighted average of the normalized mean squared error on energy, force and stress, where the weights are 0.5, 0.4 and 0.1, respectively. The models are ordered in approximately increasing complexity. The GPn correspond to new functional forms developed with POET by seeding with the interatomic potentials of Sutton Chen, GP1, GP2, or GP3. The models SC4-c, SC5-c, GP1-c, GP2-c, and GP3-c were developed by optimizing the parameters of the corresponding functional forms using the CMA-ES and the conjugate gradient optimizer..... 79

Figure 22. Average of normalized errors across validation properties. The validation metrics considered on this plot are: MAE of energies, MAE of forces, MAE of stresses, MAPE of C11, C12, C44, MAPE of 7 phonon frequencies, absolute percent error of vacancy formation energy, absolute percent error of vacancy migration energy, absolute percent error of dumbbell formation energy, MAPE of 13 low-index surface energies (except for GP3), absolute percent error of intrinsic stacking fault energy, absolute percent error of unstable stacking fault energy, absolute percent error of hcp formation energy, absolute percent error of bcc formation energy, absolute percent error of fcc lattice parameter, and absolute percent error of bcc lattice parameter. The normalization was done using min-max scaling (x-min(x))/(max(x)-min(x)) 81

Figure 23. Mean absolute errors (MAE) of GP1-c models on energies in meV/atom. The training points are red squares, and the validation points are blue circles. The first

value on top of the plot is the training MAE, and the second is the validation MAE. 84
Figure 24. Mean absolute errors (MAE) of GP2-c models on energies in meV/atom. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 84
Figure 25. Mean absolute errors (MAE) of GP3-c models on energies in meV/atom. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 85
Figure 26. Mean absolute errors (MAE) of GPn models on energies in meV/atom. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 85
Figure 27. Mean absolute errors (MAE) of GP1-c models on the components of the forces in meV/Å. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 86
Figure 28. Mean absolute errors (MAE) of GP2-c models on the components of the forces in meV/Å. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 86
Figure 29. Mean absolute errors (MAE) of GP3-c models on the components of the forces in meV/Å. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 87
Figure 30. Mean absolute errors (MAE) of GPn models on the components of the forces in meV/Å. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 87

Figure 31. Mean absolute errors (MAE) of GP1-c models on the components of the virial stress tensor in GPa. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 88
Figure 32. Mean absolute errors (MAE) of GP2-c models on the components of the virial stress tensor in GPa. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 88
Figure 33. Mean absolute errors (MAE) of GP3-c models on the components of the virial stress tensor in GPa. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 89
Figure 34. Mean absolute errors (MAE) of GPn models on the components of the virial stress tensor in GPa. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE. 89
Figure 35. Box plots of absolute errors on the validation energies. The orange lines are the median, and the values of the medians are shown on top. The boxes show the interquartile range (IQR), and the whiskers are at 1.5 of the IQR, the blue points are the outliers beyond 1.5 of the IQR. 91
Figure 36. Box plots of absolute errors on the validation components of the force. The orange lines are the median, and the values of the medians are shown on top. The boxes show the interquartile range (IQR), and the whiskers are at 1.5 of the IQR, the blue points are the outliers beyond 1.5 of the IQR. 92
Figure 37. Box plots of absolute errors on the validation components of the virial stress tensor. The orange lines are the median, and the values of the medians are shown on top. The boxes show the interquartile range (IQR), and the whiskers are at 1.5 of the IQR, the blue points are the outliers beyond 1.5 of the IQR. 93
Figure 38. Average of normalized errors across validation properties for (a) GPn models, and for (b) SC4-c models. The validation metrics considered on this plot are: MAE of energies, MAE of forces, MAE of stresses, MAPE of C_{11}, C_{12} and C_{44}, MAPE of 7

phonon frequencies, absolute percent error of vacancy formation energy, absolute percent error of vacancy migration energy, absolute percent error of dumbbell formation energy, MAPE of 13 low-index surface energies, absolute percent error of intrinsic stacking fault energy, absolute percent error of unstable stacking fault energy, absolute percent error of hcp formation energy, absolute percent error of bcc formation energy, absolute percent error of fcc lattice parameter, and absolute percent error of bcc lattice parameter. The normalization was done using min-max scaling (x-min(x))/(max(x)-min(x)) 97

Figure 39. Average of normalized errors across validation properties for the models (a) GP1-c, (b) GP3-c, (c) GP2-c, and (d) SC5-c. The validation metrics considered on this plot are: MAE of energies, MAE of forces, MAE of stresses, MAPE of C_{11}, C_{12} and C_{44}, MAPE of 7 phonon frequencies, absolute percent error of vacancy formation energy, absolute percent error of vacancy migration energy, absolute percent error of dumbbell formation energy, MAPE of 13 low-index surface energies, absolute percent error of intrinsic stacking fault energy, absolute percent error of unstable stacking fault energy, absolute percent error of hcp formation energy, absolute percent error of bcc formation energy, absolute percent error of fcc lattice parameter, and absolute percent error of bcc lattice parameter. The normalization was done using min-max scaling (x-min(x))/(max(x)-min(x)) 98

Figure 40. Vacancy formation energy (E_v) predicted by GPn models compared to DFT E_v, maximum experimental E_v, and minimum experimental E_v. See Table 23 for a list of references of the experimental values. 99

Figure 41. (a) Complexity (i.e., number of nodes) of each of the model types. The number of nodes of SC4-c and SC5-c are the same. The number of nodes of the models are 15, 19, 24, and 26 nodes for SC4-c, GP1-c, GP3-c, and GP2-c, respectively. The average number of nodes of GPn models is 18, and the average for literature models is 348. (b) Complexity (i.e., number of nodes) of the GPn models for each element. 101

Figure 42. (a) Pareto frontier of EAM-type interatomic potential models for Ni considering the absolute error on the vacancy migration energy and the number of nodes (complexity). No model has less error and is simpler than a model in the Pareto

frontier. The models SC4-c, GP1-c, GPn and Kelton_Ni belong to the frontier. The references are: Chen_Ni [189], Daw_Ni [144], Wolfer_Ni [143], and Kelton_Ni [190]. (b) Number of times that an EAM-type model belongs to the Pareto frontier divided by the number of times that the model has validation values available across the elements and properties. The metrics considered are the validation MAE on energy, force, and stress, MAPE on elastic constants, MAPE on 13-low index surface energies, absolute error on vacancy formation energy, absolute error on vacancy migration energy, absolute error on dumbbell formation energy, absolute error on intrinsic stacking fault energy, absolute error on unstable stacking fault energy, absolute error on hcp formation energy, absolute error on bcc formation energy, absolute error on fcc lattice parameter, absolute error on bcc lattice parameter, and absolute error on each high-symmetry phonon frequency: $\nu_L(X)$, $\nu_T(X)$, $\nu_L(L)$, $\nu_T(L)$, $\nu_L(K)$, $\nu_{T1}(K)$, and $\nu_{T2}(K)$. 103

Figure 43. Number of times that an EAM-type model belongs to the Pareto frontier divided by the number of times that the model has validation values available across the elements and properties, excluding GPn models to analyze the transferability of GP1-c, GP2-c, and GP3-c. The metrics considered are the validation MAE on energy, force, and stress, MAPE on elastic constants, MAPE on 13-low index surface energies, absolute error on vacancy formation energy, absolute error on vacancy migration energy, absolute error on dumbbell formation energy, absolute error on intrinsic stacking fault energy, absolute error on unstable stacking fault energy, absolute error on hcp formation energy, absolute error on bcc formation energy, absolute error on fcc lattice parameter, absolute error on bcc lattice parameter, and absolute error on each high-symmetry phonon frequency: $\nu_L(X)$, $\nu_T(X)$, $\nu_L(L)$, $\nu_T(L)$, $\nu_L(K)$, $\nu_{T1}(K)$, and $\nu_{T2}(K)$. 104

Figure 44. A summary of previous studies of elemental clusters in terms of exploring their atomic structures. We have considered literature that used DFT to find atomic structures as well as systems covered in the Cambridge Cluster Database that used empirical potentials. We have considered 55 different elements across the periodic table with size regimes from 3-55 atoms in the cluster. Note: even when the cluster of

a particular element and size has been explored in the literature, we may have discovered new clusters for that element and size. 108
Figure 45. Schematic workflow of the genetic algorithm used for the Quantum Cluster Database 112
Figure 46. Template clusters used for identifying low-energy clusters using correlations. 115
Figure 47. Pearson correlation coefficients between elements in the Quantum Cluster Database sorted to facilitate the identification of trends. 116
Figure 48. Workflow for generating the Quantum Cluster Database. 118
Figure 49. Count of the difference between the energy of the lowest-energy clusters found in the literature (minus 1 meV/atom to account for DFT precision) minus the energy of lowest-energy clusters discovered in this work. The Quantum Cluster Database work discovered 501 lowest-energy clusters that have a lower energy than the lowest-energy clusters from the literature. 124
Figure 50. The Quantum Cluster Database covers 849 regions that were previously unexplored (shown in orange). A summary of previous studies of elemental clusters in terms of exploring their atomic structures. We have considered literature that used DFT to find atomic structures as well as systems covered in the Cambridge Cluster Database that used empirical potentials. We have considered 55 different elements across the periodic table with size regimes from 3-55 atoms in the cluster. Note: even when the cluster of a particular element and size has been explored in the literature, we may have discovered new clusters for that element and size. 125
Figure 51. Interface of the Quantum Cluster Database 127

1 Introduction and Overview

The ability to design materials with specific properties is valuable for a wide variety of industries. For example, better materials can have a positive impact on the renewable energy industry because they are an essential building block of clean energy technologies. In addition to benefiting the environment, improved materials can positively impact society through better electronic or mechanic devices, among others. A challenge is that the development of a new material usually takes between 15 and 25 years from initial research to the first use. This challenge is addressed by the Materials Genome Initiative by promoting the use of data-driven methods for accelerating the development of novel materials.[1-4]

Materials informatics plays a key role in accelerating the rational design of materials, together with other essential components like experimental approaches. For example, the rate of discovery of materials has been increased through the use of computational methods to create databases of materials (e.g., databases of structure-property relationships) and the application of artificial intelligence techniques to predict properties of materials, which have informed experimental studies.[1, 3-7] The advances in materials informatics have been inspired by the many examples of success of data-driven methods in other fields, and they have been enabled by the access to greater computational resources, and by the improvements in algorithms and code libraries. Materials informatics is a relatively new field compared to informatics in other sciences, like bioinformatics and cheminformatics,[5, 7] and there are many opportunities for addressing challenges like the identification of materials with particular properties by data-driven techniques (e.g., machine learning

models and computational materials simulations) or the development of databases of materials.

Simulations and models of atomic systems at large time and length scales are important because several properties required for designing materials are associated with atomic configurations and processes at these scales. Some examples of these kinds of structures are planar defects like surfaces, that are important for catalytic properties, and grain boundaries, that are important for mechanical properties of the material. Examples of processes at large time scale is the motion of dislocations, which play an important role in determining the mechanical properties,[8-9] or ionic diffusion, which play an important role in energy materials.[10-13] In this dissertation, we demonstrate an approach for developing fast and accurate interatomic potential models for atomic simulations, using a physics-informed hypothesis space, which leads to transferrable models.

In addition, the thesis presents the development of a database of atomically precise nanoclusters. The database can inform the experimental synthesis of nanoclusters, it can be used to identify (or screen) clusters suitable for a variety of purposes that require specific properties, or to train machine learning models, among other applications.

The thesis is divided into five chapters. The first chapter introduces the work presented in the thesis and provides an overview of the dissertation. The second chapter discusses how our research group used genetic programming to develop simple, fast and accurate interatomic potential models for Cu using genetic programming. The third chapter discusses how the interatomic potential models discovered for Cu transfer well to other elemental fcc systems close to Cu on the periodic table. The fourth chapter discusses the development of the Quantum Cluster Database, a database of atomically precise

nanoclusters. The fifth chapter concludes the work presented on this work and discusses future directions.

2 Fast, Accurate, and Transferrable Many-Body Interatomic Potentials by Symbolic regression

2.1 Background

2.1.1 The Potential Energy Surface

Atomistic simulations and models of materials (and molecules) would ideally determine the state and behavior of a system by analytically solving the Schrödinger equation. However, it is impossible to solve it analytically for most real-world systems due to the computational cost, and approximations are required. A commonly used approximation is the Born-Oppenheimer approximation, which leads to the potential energy surface. The potential energy surface allows researchers to perform atomic-scale computational research through the calculation of free energies and other thermodynamically averaged properties, the simulation of the evolution of a system over time, the identification of stable and metastable atomic configurations, the calculation of vibrational modes and frequencies, among many others. The potential energy surface can be approximated using *ab initio* methods, like density functional theory, or methods that do not explicitly consider electrons, like interatomic potential models. In this section, we will go through some of the steps that lead to the potential energy surface starting from the time-independent Schrödinger equation:

$$\hat{H}\psi(\mathbf{r}) = E\psi(\mathbf{r}) \tag{2.1}$$

where $\hat{H}$ is the Hamiltonian operator, ψ is the wavefunction, $\mathbf{r}$ are the coordinates of the particles in the system, and E is the ground state energy of the system or the energy of an

excited state. The time-independent Schrödinger equation can be written using the many-body Hamiltonian:

$$\hat{H} = \hat{T} + \hat{V} \tag{2.2}$$

where $\hat{T}$ and $\hat{V}$ are the kinetic and potential energy terms, respectively. The atomic systems of interest are composed of nuclei and electrons. The kinetic energy term is the sum of the kinetic energy of nuclei and electrons:

$$\hat{T} = -\sum_{i}^{N_e} \frac{\hbar}{2m_e} \nabla_i^2 - \sum_{A}^{N_n} \frac{\hbar}{2m_A} \nabla_A^2 \tag{2.3}$$

where N_e is the number of electrons, N_n is the number of nuclei in the system, m_e is the mass of the electron, and m_A is the mass of a nucleus. The potential energy term can be written as the addition of the Coulombic interactions between nuclei, between electrons and nuclei, and between electrons:

$$\hat{V} = \hat{V}_{n-n} + \hat{V}_{e-n} + \hat{V}_{e-e} \tag{2.4}$$

Equivalently:

$$\hat{V} = \sum_{A<B}^{N_n} q_e^2 \frac{Z_A Z_B}{|\mathbf{R_A} - \mathbf{R_B}|} + \sum_{i}^{N_e} \sum_{A}^{N_n} q_e^2 \frac{-Z_A}{|\mathbf{r_i} - \mathbf{R_A}|} + \sum_{i<j}^{N_e} q_e^2 \frac{1}{|\mathbf{r_i} - \mathbf{r_j}|} \tag{2.5}$$

where Z_A is the atomic number of the nucleus, $\mathbf{r_i}$ represents the position of electron "i" and $\mathbf{R_A}$ gives the position of nucleus "A".

The Born-Oppenheimer approximation assumes that the electrons and nuclei are not coupled, and the nuclear kinetic energy can be ignored. A fundamental argument for these

assumptions is that the mass of the nuclei are much greater than the mass of the electrons, $m_e \ll m_A$, so that the electrons adjust adiabatically to changes in nuclear coordinates; electrons move much faster than nuclei. The Born-Oppenheimer approximation leads to the electronic Schrödinger equation:

$$\left(-\sum_{i}^{N_e} \frac{\hbar}{2m_e} \nabla_i^2 + \sum_{i}^{N_e} \sum_{A}^{N_n} q_e^2 \frac{-Z_A}{|\mathbf{r_i} - \mathbf{R_A}|} + \sum_{i<j}^{N_e} q_e^2 \frac{1}{|\mathbf{r_i} - \mathbf{r_j}|} \right) \psi(\mathbf{r}, \mathbf{R}) + \sum_{A<B}^{N_n} q_e^2 \frac{Z_A Z_B}{|\mathbf{R_A} - \mathbf{R_B}|} = E_e(\mathbf{R}) \psi(\mathbf{r}, \mathbf{R}) \tag{2.6}$$

where $E_e(\mathbf{R})$ is the electronic energy eigenvalue as a function of the nuclear coordinates, and the nuclear coordinates are treated as constant parameters. The potential energy surface can be obtained by solving the electronic Schrödinger equation, equation (2.6), at various nuclear coordinates $\mathbf{R}$.

2.1.2 Interatomic potential models

The potential energy surface is most accurately represented by quantum mechanical calculations. However, exact solutions to Schrödinger equation are not available for most real-world systems, and the computational methods used to approximate solutions to Schrödinger's equation are limited by their computational cost and how the cost scales with the number of particles in the system. Density functional theory (DFT)[14-15] is one of the most widely used methods for approximating solutions to Schrödinger's equation. The wide acceptance of DFT is due in large part to its favorable trade-off between speed and efficiency, it scales as $O(n^3)$ with the number of symmetrically distinct electrons in the system, for large systems.[16-17] The cubic scaling limits the applicability of DFT for

statistical sampling of thermodynamic averages, to calculate the properties of systems containing more than about 1000 symmetrically distinct atoms, or for more molecular dynamics simulations more than about 10 nano seconds long. In practice, most DFT calculations are used for significantly smaller time and length scales. However, DFT is still among the best *ab initio* methods in terms of the tradeoff of computational cost and accuracy; other quantum mechanical methods generally scale more poorly with system size[18] and/or have a large prefactor that offsets otherwise favorable scaling.[19-20]

2.1.2.1 Empirical interatomic potential models

Ab initio approaches are derived from fundamental physical principles and have good transferability across many chemical compositions and types of materials, and their accuracy is also good across a wide variety of systems. However, many atomic configurations and processes of practical, technological, or scientific interest exist on time and length scales that are not accessible using quantum mechanics. To gain access to greater scales, researchers have developed alternative approaches for calculating the potential energy surface. An example of such methods are the interatomic potential models (or force fields), which express the potential energy surface as a function of the nuclear coordinates and usually scale as $O(n)$ with the number of atoms in the system. Some interatomic potential models are derived from physical principles, they are known as empirical (or classical) interatomic potential models. Two examples of empirical interatomic potential models are the Coulomb potential,[21] in which the nuclei are treated as point charges that interact electrostatically, and the Lennard-Jones interatomic potential (equation (2.7)),[22] which reproduces the r^{-6} decay in the dispersion interaction at large distances. The Lennard-Jones interatomic potential model has the form:

$$E_{tot} = \frac{1}{2}\sum_{ij} V_{LJ}(r) \tag{2.7}$$

where E_{tot} is the total energy of the atomic system, the first summation goes over each atom, “i", in the system, the second summation goes over each neighbor, “j”, of an atom “i" within a cutoff distance, and V_{LJ} is the Lennard-Jones pair interaction term. Usually, V_{LJ} is multiplied by a smoothing function to avoid the introduction of discontinuities to the potential energy surface due to the cutoff distance. The pairwise term is:

$$V_{LJ} = 4\varepsilon\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right) \tag{2.8}$$

where r is the interatomic distance between atoms “i" and “j”, ε is the well depth of the potential energy, and σ is the interatomic distance at which V_{LJ} is zero. Other examples of empirical interatomic potential models derived from physical principles include the Buckingham potential,[23] in which the Lennard-Jones potential was adjusted to produce exponential repulsion at short distances, the Stillinger-Weber potential,[24] designed to reproduce the equilibrium bond angle in crystalline silicon, and the embedded atom potential (Equation (2.9)). [25] Conceptually, the embedded atom method potential states that metals are embedded in a sea of delocalized electrons, and it can be justified using a tight-binding model.[26] The embedded atom method model is:

$$E_{tot} = \frac{1}{2}\sum_{ij} V(r) + \sum_{i} F(\bar{\rho}_i) \tag{2.9}$$

where E_{tot} is the total energy of the atomic system, $V(r)$ is a pair interaction term that depends on the interatomic distance, r, between atoms “i" and “j”, F is the embedding function, and $\bar{\rho}_i$ is the electron density at site “i".

$$\bar{\rho}_i = \sum_{i \neq j} \rho(r) \tag{2.10}$$

where ρ is the electron density function (also known as the electron transfer function).

Other empirical interatomic potential models were derived based on the concept of bond order, known as bond-order potentials; Brenner showed that they are closely related to potentials in the embedded atom family.[27] To account for the covalent component of bonding, bond angles were introduced and gave origin the modified embedded atom method,[28-29] the angular dependent potential,[30] and the bond-order potentials such as the Tersoff potential.[31] More advanced approaches, such as the Charge Optimized Many Body potential (COMB), incorporate the equilibration of charge in the model.[32-34] Other techniques combine several of the empirical interatomic potentials into a single model, such as the ReaxFF force field.[35-36] Reviews of these and other potentials can be found in references [8, 26, 37].

These physics-derived interatomic potential models are often used to model systems at particularly large time or length scales. The fastest interatomic potential models are empirical potentials that are as simple as the Lennard-Jones potential or the embedded atom method potential. These are widely used to simulate materials at extreme time and length scales because they can evaluate energies at speeds of ~1 μs / atom on a CPU core and they scale linearly with the number of atoms in the system.[8] The physics-derived interatomic

potential models can generalize reasonably well to a variety of systems due to their foundation on physical concepts. However, their accuracy is limited by their functional forms, which are typically manually constructed and generally need to be parameterized for each system. A challenge with the most accurate models, like COMB and ReaxFF, is that they may contain hundreds of parameters and it is difficult to optimize them.[38-40] There is active research in determining how machine learning can be used to systematically find good sets of parameters for such models.[41-46]

2.1.2.2 Supervised machine learning for developing interatomic potential models

A commonly known observation, known as Moore's law, is that the number of transistors in an integrated circuit doubles every two years, and the computer chip performance roughly doubles every 18 months. With the growing access to computational resources and algorithmic advances, it has become more practical to generate data using *ab initio* calculations and to use that data to fit interatomic potential models through machine learning methods. In the past two decades, researchers have made great progress in developing methods for fitting interatomic potential models to data generated using DFT or other quantum mechanical methods. As initially demonstrated by Eroclessi and Adams, a single DFT calculation provides as training data not only the energy of a given configuration of atoms but also, with little extra computational cost, the forces on each of the atoms, providing 3N+1 points of training data for a single DFT calculation on a system with N symmetrically distinct atoms.[47] In recent years, the components of the virial stress tensor have been used on the training set, adding 6 datapoints per DFT calculation.

Initially the approach of fitting interatomic potentials to *ab initio* data was primarily used to parameterize potential models with manually-constructed functional forms, such as empirical potentials, but then the use of machine learning to fit DFT data shifted the paradigm towards functional forms determined by the machine learning methods utilized rather than expressions from underlying physical interactions. This is a supervised machine learning approach to developing interatomic potential models.[4] Supervised learning has the goal of determining a function f that makes accurate predictions of a value y for sets of input data $\mathbf{x}$. In the context of interatomic potential models, $\mathbf{x}$ represents the atomic species and nuclear coordinates, y is the value on the potential energy surface, and f is the learned interatomic potential model. Supervised machine learning requires three fundamental steps.

The determination of the hypothesis space is the first step. The hypothesis space is the space of mathematical expressions that will be searched to find f. If left unconstrained, the space of all possible functions contains an infinite number of functions that perfectly reproduce the training data, so it is necessary to place constraints on the hypothesis space. Many of these functions will over-fit the data especially if they are complex. An example of how to constrain the hypothesis space is to include only functions spanned by a small, finite set of basis functions, or to penalize functions that are unlikely to have high predictive power (e.g. because they are too complex or physically unlikely). The penalty can be considered through regularization; sometimes done by assigning another term to the objective function, which leads to the next step.

The second step is to determine an objective function that determines how good each of the models in the hypothesis space is. A common objective function that is optimized when

fitting potential models is the mean squared error in the prediction of normalized energies, forces, and/or virial stresses relative to the training data. The third and final step is to select a technique for searching the hypothesis space for good functions as determined using the objective function. For example, if the hypothesis space consists of neural networks, the backpropagation algorithm[48] can be used to find the best set of weights. For a hypothesis space that is a linear combination of basis functions, then linear algebra can often be used to identify the best function.

During the development of interatomic potential models using supervised machine learning, the hypothesis space is usually constrained through two physically-motivated conditions. The first constraint is that interactions between atoms or groups of atoms are limited to a certain cutoff radius. Through this assumption, atoms that are separated by a distance larger than the cutoff radius do not interact. This is reasonable for most systems, because it suggests that the local contributions have a much greater impact than the contributions from atoms at large distances. The cutoff radius is usually about 4-9 Å[49-50] from a central atom. In cases where Coulomb or dispersion interactions are important, these can be added to the short-range potential.[51-53]. The second physically motivated constraint on the hypothesis space is that the potential energy must be invariant to isometric transformations of the system like rotation, reflection, translation, and combinations thereof.

In the development of interatomic potential models through supervised machine learning, the two physically-motivated constraints on the hypothesis space described above are commonly implemented through descriptors (or fingerprints) of the local atomic environment around an atom within a cutoff distance; where every atom has a descriptor

associated with it. The use of machine learning for building interatomic potential models has greatly benefited from the development of robust descriptors of the local atomic environment. Generating robust fingerprints can be challenging in part because they need to be invariant to isometries of the system. [54-67] Using the descriptors, the machine learning algorithm tries to find a function of these descriptors that optimizes the objective function, which may be represented as equation (2.11). There are several descriptors that have been widely studied, including atom-centered symmetry functions,[56] bispectrum components,[57] Coulomb matrices,[60] and the smooth overlap of atomic positions (SOAP).[68] Such descriptors are commonly used in a variant of machine learning approaches including neural network potentials (through the Behler-Parinello approach),[56] Gaussian approximation potentials (GAP),[57] and spectral neighbor analysis potentials (SNAP).[58-59] Several of these descriptors and approaches for constructing interatomic potentials are well-covered literature reviews. [50, 68-70] Equation (2.11) is a way of representing the best model $\hat{f}$ in the case when the mean squared error is used as the objective function.

$$\hat{f} = \arg\min \frac{1}{n} \sum_{i}^{n} (f(\mathbf{x}_i) - y_i)^2 \tag{2.11}$$

where "y" is the target value in the potential energy surface, "f" is a prospective model, and **x** is the descriptor of the atomic environment.

Two other machine learning techniques that have emerged in recent years are moment tensor potentials [71] and graph network potentials [72-79]. These approaches do not implement the descriptors of the atomic environment described above. The potential energy in moment tensor potentials is represented as a linear combination of polynomial basis functions that account for one-body, two-body, three-body, or higher, interactions. Cluster expansions

and moment tensor potentials are related because they represent the energy as a many-body expansion (i.e., a linear combination of *k*-body interactions), but moment tensor potentials define the basis over continuous atomic coordinates instead of discrete atomic sites. An extension to the cluster expansion that generates a continuous potential energy surface was recently introduced by Drautz [62], creating a new class of interatomic potential models called atomic cluster expansions that, like moment tensor potentials, achieve linear scaling with the number of neighboring atoms by transforming a sum over products into a product over sums; [62] Seko et al. have presented a similar approach.[63]

On the other hand, graph network potentials usually represent the atomic structure as a graph data structure, where the atoms are the nodes (or vertices), and the bonds are the edges. In this way, information about the atomic structure is encoded into the graph network by the connectivity of the nodes, with distances between atoms provided as input values. The early research projects related to using graph networks for atomic-scale modeling mainly focused on applications to molecules,[80] but recently there have been successful applications of this approach to crystalline materials as well. [73-75, 79, 81] Gilmer et al. have pointed out that many of the current implementations of graph networks can be expressed in a common message-passing framework, in which information is iteratively passed from node to node along edges connecting the nodes.[76] More information about the graph network potentials and the moment tensor potentials can be found on [82]. In this dissertation, we discuss a third approach for developing interatomic potential models using machine learning via genetic programming.

2.1.3 Symbolic regression via genetic programming

Symbolic regression is a supervised learning technique that solves the problem of finding a functional form that fits a given dataset. Historically, a goal of many scientists and engineers has been to identify natural laws or empirical relationships in the form of mathematical expressions, and this is one of the drivers for research in symbolic regression. [83-96] Symbolic regression can be implemented through genetic programming, a method based on Darwinian evolution. [97] In the genetic programming approach to symbolic regression, mathematical expressions are usually represented as tree graphs, and these trees are called individuals (Figure 1). Genetic programming maintains a set (or sets) of trees and evolves them using crossover and mutation operations to identify better functions; the set of individuals is called a population. Conceptually, the optimization engine of this kind of symbolic regression is a genetic algorithm[98-99]. The genetic programming algorithm chooses trees for crossover and mutation based on how well they fit the data with the expectation that the offspring will inherit qualities associated with the good fitness. [100] An example of a crossover operation is to select two trees and replace a branch of one tree with the branch of another tree (Figure 1). Mutation provides a way to explore the global landscape of functions. [97]

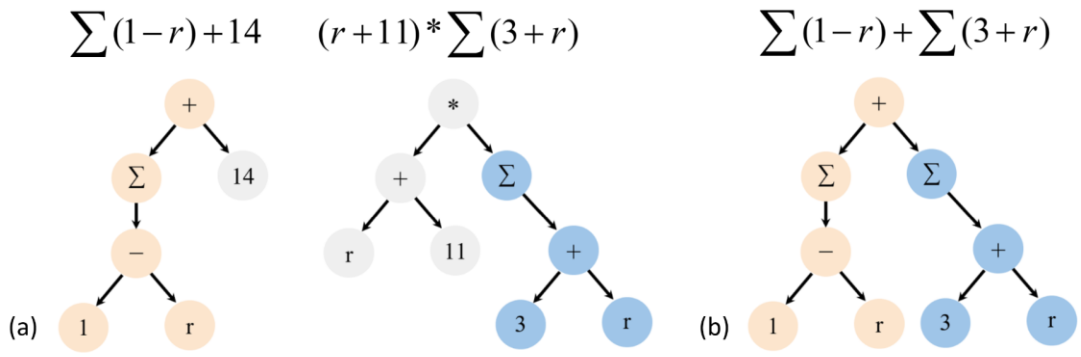


Figure 1. Crossover operation in genetic programming. (a) Trees selected for crossover. (b) Offspring after the crossover operation.

2.2 Introduction

About 25 years ago, Ercolessi and Adams[101] demonstrated a new approach for fitting interatomic potential models to *ab initio* calculations, and in this way, they expanded the number of datapoints available for developing interatomic potentials. Researchers progressively started implementing their approach, and in the past 10 years it has been a fundamental component for the development of interatomic potential models using machine learning. [50, 53-54, 56-58, 60, 64-65, 69, 71, 94, 102-109] This is a supervised learning approach for developing interatomic potential models, [4] in which an optimization algorithm is used to search a hypothesis space of possible functions to find those that best reproduce the energies, forces, and possibly other properties of a set of training data. The interatomic potential models developed through this technique usually achieve an accuracy like the accuracy of the *ab initio* method used to generate the training data, with the advantage of having linear scaling and lower computational cost than quantum mechanical methods.

Another approach to developing interatomic potential models is to use fundamental physical relationships to derive a parameterized function. The parameters of this function are typically then fit to a smaller set of training data, compared to the supervised learning approach. These are known as classical or empirical interatomic potential models. Some potential models generated using this approach include the embedded atom method (EAM) and bond-order potentials.[31, 110-115]. The models developed in this way scale linearly with the number of atoms in the system and are orders of magnitude faster than first principles methods.

Both techniques for developing interatomic potential models have advantages and disadvantages. The machine learning approach can be used to develop models for a wide variety of different chemical systems, and because many machine learning algorithms explore a large hypothesis space, they are usually able to achieve high levels of accuracy on structures where the local environments of the atoms are similar to those that are contained in the data used to train the model.[56-58] The interatomic potential models developed through the classical approach and from fundamental physical relationships are often simpler and orders of magnitude faster than machine learning potential models,[8] allowing them to be used to model systems at much longer time and length scales. Because they are derived from physics, they can be expected to perform relatively well when they encounter local environments that are unlike the ones they were trained on. Another advantage to the latter approach is that the hypothesis space of these potential models is relatively small compared to most machine learning potentials, meaning that less data is required to train them but also that they are typically unable to achieve the same level of accuracy as many potentials generated using machine learning.

In this thesis, we present a hybrid approach based on symbolic regression performed using genetic programming, in which simple mathematical expressions for the potential energy surface are identified and optimized by simulating the process of natural selection.[100, 116] Genetic programming has been used in the past for a variety of scientific and engineering applications.[97] For example, it has been used to rediscover fundamental physical laws[83] and it has been applied in materials science to find descriptors of complex material properties.[90, 117] It has also previously been used to search for simple two- and three-body interatomic potentials.[87, 91-92, 94] Here we go beyond these previous efforts by demonstrating that genetic programming is capable of finding fast, accurate and transferable many-body potentials for a metallic system from *ab-initio* calculations.

We constructed a physically meaningful hypothesis space by analyzing simple interatomic potential models that were derived from physical principles.[113-115] Many of these empirical interatomic potential models present similarities in their functional forms. The hypothesis space that we built takes advantage of this fact and contains many of these models. This space contains a wide variety of potential models derived from fundamental physical interactions, including nearly all pair potentials (e.g. Lennard-Jones,[118] Coulomb,[21] Morse[119]) as well as many-body glue potentials,[113] bond-order potentials (without the bond angle terms),[27, 31, 113, 120] and combinations thereof. The construction of this hypothesis space is an essential component to our approach.

The hypothesis space that we designed consists of all functions that can be constructed from combinations of addition, subtraction, multiplication, division, and power operators; constant values and distances between atoms; and an operator that performs a sum over functions of distances between a given atom and all neighbors within a given cutoff radius.

Even for relatively simple hypothesis spaces such as this one, it is difficult to enumerate a list of even relatively simple functional forms that can be created due to the large number of ways in which the various operators and values can be combined.[90] Here we use a genetic algorithm and multi-objective optimization to search this hypothesis space for interatomic potentials that are simple (and thus more likely to be generalizable [121]), fast, and accurate. Additional details of our approach are provided in the Methods section.

2.3 Methods

2.3.1 The hypothesis space of the supervised learning problem

We built a machine learning algorithm that uses genetic programming to search a hypothesis space of models that can be constructed by combining real numbers, addition, subtraction, multiplication, division, exponentiation, and a sum over neighbors of an atom. The motivation for creating this hypothesis space came from the observation that many physics-derived interatomic potential models use similar functions and operators. In this way, the hypothesis space encodes physics information which is important for having a robust model. Within this hypothesis space, each function can be represented as a tree graph, as shown in Figure 2. The space was constrained so that the maximum number of summations over neighbors was 6, no nested summations over neighbors were allowed, the maximum allowed depth of a tree was 32 and the maximum allowed number of nodes was 511. To ensure smoothness of the potential energy surface as represented by the model, all the sums over neighbors are multiplied by the following smoothing function before the sum over neighbors is taken:[122]

$$f(r) = \left(2r^2 - 3r_{in}^2 + r_{out}^2\right)\left(r_{out}^2 - r^2\right)^2 \left(r_{out}^2 - r_{in}^2\right)^{-3} \quad (2.12)$$

where r_{in} and r_{out} are the inner and outer cutoff radii, for GP1 and GP2, r_{in} = 3 Å and r_{out} = 5 Å, including the 3rd nearest neighbors.[123]

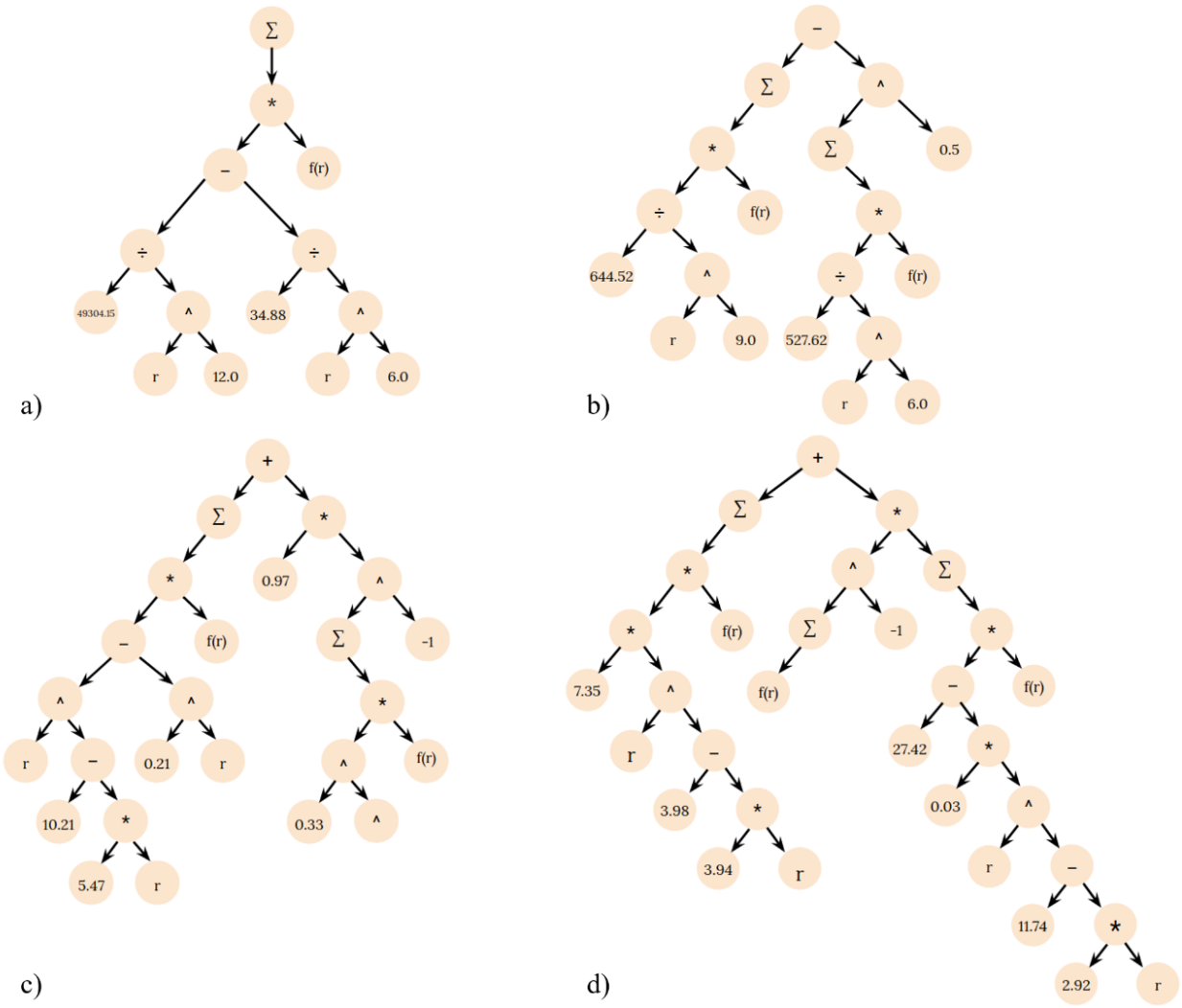


Figure 2. Tree graphs of a) Lennard-Jones potential parametrized for argon, equation (2.14) , b) Sutton-Chen EAM potential parametrized for copper, equation (2.16), c) GP1 and d) GP2

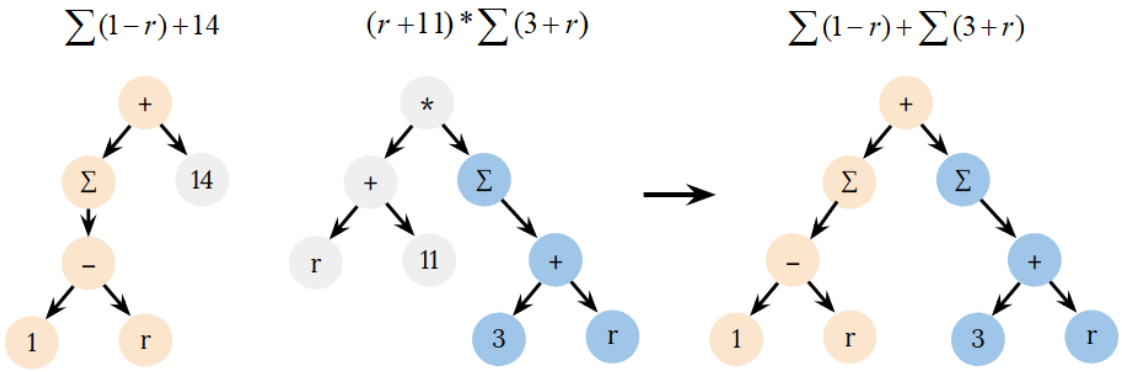


Figure 3. Example of a crossover operation

2.3.2 Description of the artificial intelligence algorithm

Genetic programming evolves computer programs following Darwin's natural selection by performing crossover and mutation operations on a set of individuals. Crossover was performed by 2 different operations: by randomly selecting a branch from one tree and replacing it with a randomly selected branch of another tree (Figure 3), and by creating a linear combination of 2 randomly selected branches from 2 different tress – the first method was randomly selected 90% of the times and the second one 10% of the times. The mutation operation performed 3 different sub-operations with equal probability: crossover of a tree with a randomly generated tree, swapping the arguments of a binary non-commutative function, and slightly modifying the expression tree by replacing (or inserting) a randomly selected non-terminal node with a randomly selected operator.[97] The randomly generated trees were generated with the grow or full method with equal probability,[100] and the depth was drawn from a Gaussian distribution of mean 5 and standard deviation of 1. The overall algorithm performed crossover with a probability of 0.9, and mutation with a probability of 0.1.

Increasing diversity is known to improve the quality of the optimization.[97] To increase diversity, we implemented a hierarchical way of creating separate environments in which the individuals (i.e., potential models) evolved. We ran the algorithm on 12 processors, and each processor had its own environment, consisting of a population of models and a subset of the training data. Conceptually this allows potentials within a specific environment to develop characteristics that are unique, increasing the diversity. Candidates for crossover and mutation were selected from 3 different sets of models with equal probability:

(1) The population of the current processor. Every 20,000 crossover and mutation operations, 100 individuals were selected based on their fitness (equation (2.18)) with Pareto tournament selection of size 10 while the rest were discarded.[88, 124-125]

(2) A global set of models. Each processor tried to add the 100 individuals selected in part (1) to the global subset every 20,000 crossover and mutation operations. The models on the global set were then evaluated based on speed (to model large time and length scales), fitness (for accurate results), and complexity (for generalizability). The speed of each model was estimated by the number of summations over neighbors. The complexity was evaluated by the number of nodes in the tree graph. To identify the best models, we generated separate convex hulls with respect to fitness and complexity for each number of summations (speed) in a potential. Only the models on these convex hulls were retained in the global set.

(3) Individuals from other processors. Each processor was allowed to communicate with other processors every 5000 crossover and mutation operations, importing the current set of individuals from them.

Selection with equal probability was performed when getting an individual from the global set. Tournament selection of size 10 was used for getting individuals from the population of the current processor and from the populations of other processors.

The training data was also arranged in hierarchical subsets to increase diversity and reduce the speed of evaluating fitness. Globally, a subset of 75 energies, 75 forces, and 75 stresses was randomly sampled from the full set of training data every 20,000 crossover and mutation operations. The fitness of the global set of models was evaluated using this subset of training data. The training data on each processor (15-30 energies, forces and stresses) were randomly selected from the global subset of the training data, and this local subset was used to evaluate fitness locally on each processor. The subset of training data for each processor was selected from the global subset because individuals that migrate from a processor to the global set are more likely to survive if the environment is similar.

Optimization of potential model parameters was performed using the covariance matrix adaptation evolution strategy (CMA-ES) optimizer and a conjugate gradient (CG) optimizer.[126-127] The CMA-ES algorithm was selected because it performs well in nonlinear or non-convex problems. The potential models on the global set of best individuals were optimized with the CMA-ES every 10,000 crossover and mutation operations by one processor. In contrast, the CG algorithm performed one optimization step for every individual generated by crossover or mutation.

The genetic programming algorithm took about 330 CPU-hours to find the exact Lennard Jones potential, 3600 CPU-hours to find the exact Sutton Chen potential, and 360 CPU-hours to find GP1, GP2 and GP3. We note that it is likely that with additional tuning and performance enhancements the efficiency of the algorithm can be improved. To facilitate

this, our code is open source and available at https://gitlab.com/muellergroup/poet. We named our code Potential Optimization by Evolutionary Techniques (POET).

2.3.3 Details about the target data

The DFT data were computed using the Vienna Ab initio Simulation Package[128] (VASP) with the Perdew-Burke-Ernzerhof [129] (PBE) generalized gradient approximation (GGA) exchange correlation functional. The projector augmented wave method [130] (PAW) Cu_pv pseudopotential was used for copper. Efficient *k*-point grids were obtained from the *k*-point grid server with MINDISTANCE = 50Å. [131] A cutoff energy of 750 eV and ADDGRID = TRUE in VASP were required to converge the stress tensor to less than 0.05 GPa. The elastic constants were converged to within 3 |% error| using a MINDISTANCE = 100Å. The DFT point defect energies were computed by linear extrapolation. [132] The phonon dispersion curves were computed on a 3×3×3 supercell. The DFT calculation used a 5×5×5 *k*-point grid and electronic self-consistency convergence of 10^{-8} eV. The radial distribution function molecular dynamics simulations were performed in the NVT ensemble at the experimental 1400K liquid density on a 3×3×3 supercell. The temperature was increased from 300K to 2500K during 1 ps. Then the temperature was maintained at 2500K during 10 ps. Then, the temperature was decreased from 2500K to 1400K over 1ps. Then the temperature was maintained at 1400K during 1ps. Finally, the radial distribution function data was collected at 1400K over 40 ps. The DFT molecular dynamics for the radial distribution function was performed with a cutoff energy of 400 eV for the equilibration steps and 750 eV for the final 40 ps during which data was collected. Electronic self-consistency convergence was 10^{-5} eV and only the *k*-point at Γ was used. For the computation of the fitness of the models, the energies were transformed by subtracting the

minimum and dividing by the standard deviation, and the forces and stresses were standardized by subtracting the mean and dividing by the standard deviation.

The data used to rediscover the Lennard-Jones potential and the SC potential, and the data used to validate GP1 and GP2 were computed on LAMMPS. Instructions required to use GP1 and GP2 on LAMMPS are provided on the Methods section, and the required files can be provided upon request. Lennard Jones calculations used a cutoff distance of 7.5 Å, and SC, GP1, GP2 and GP3 calculations used a cutoff distance of 5 Å.

The dumbbell point defect was formed by displacing an atom along a <100> direction on an fcc structure. The relaxed vacancy formation energy, the migration energy, the activation energy, and the dumbbell formation energy of GP1, GP2, GP3, EAM2 and EAM1, were computed with 6x6x6 supercells, CuNi used a supercell with 1200 atoms, Cuu3 500 atoms, and Cuu6 250 atoms. The DFT target values were obtained by linear extrapolation of the values at 2×2×2 and 3×3×3 supercells with respect to the inverse of the supercell size.[132] The dumbbell formation energies of ABCHM, Cu1, Cu2, and MCu31 were computed with a 3×3×3 supercell.

We used the weighted surface energy as a target property for validation of GP1 and GP2, it is defined as: [133]

$$\bar{\gamma} = \frac{\sum_{\{hkl\}} \gamma_{hkl} A_{hkl}}{\sum A_{hkl}} \tag{2.13}$$

Here, γ is the surface energy, A_{hkl} is the total area of all the planes in the *hkl* family in the Wulff construction.[134] The % error of the weighted surface energy was computed by comparing the experimental (except for GP1 and GP2) target value reported in the paper

against the weighted surface energy predicted by the potential.[133, 135] The target value of GP1 and GP2 was the weighted surface energy computed by DFT.

The DFT, GP1, GP2 and GP3 intrinsic stacking fault energy and unstable stacking fault energy were computed with a (111) slab, with a gap between periodic slabs of 20 Å and a thickness of 22 (111) atomic layers. The atoms were only allowed to relax in the direction normal to the slab interface for the USF computation. The intrinsic stacking fault and the unstable stacking fault were formed by displacing atoms above a $\{111\}$ plane along a $\langle 211 \rangle$ direction by $a_0/6$, and $a_0/12$, respectively.[136]

2.3.4 Details about the interatomic potential models from the literature

Table 1 shows the acronyms of the interatomic potential models mentioned in this chapter. The models from the literature used three types of sources data for the fitting procedure: experimental data, *ab initio* data, or a combination of both. ABCHM, Cu1, Cu2, MCu31 and EAM1 used experimental and *ab initio* data for training. EAM1 used *initio* data for relative energies of hcp, bcc, and fcc phases, and energies of the fcc phase and a diatomic molecule under strong compressions, and experimental data for other properties. About 40 % of the training data of ABCHM, Cu1, Cu2 and MCu31 was *ab initio* data; it included relative energies between phases, lattice parameters, vacancy formation energy and interstitial formation energy.

The accuracy (e.g., percent errors) reported on this chapter for each of the interatomic potentials are obtained directly from the original paper in which the potential was first published, with one exception: the weighted surface energies of CuNi, EAM1 and Cuu3

were computed using the interatomic potentials from the Interatomic Potentials Repository using LAMMPS.[37, 137]

Table 1. Acronyms used for the interatomic potential models.

Acronym	Description	Fitting procedure
SC[138]	Sutton-Chen (SC) EAM interatomic potential. Defined by a Finnis-Sinclair[111] potential with a van der Waals term for long-range interactions. Developed for metallic bonding and mechanical interactions between clusters	Experimental data
GP1, GP2 and GP3	Interatomic potential developed as part of this thesis	*Ab initio* data
EAM2[123]	Compared against EAM1 in the article.[123] Defined by a Morse[119] function, the universal equation of state[139] and the density related to a 4s orbital. Developed for energies and mechanical stability of non-equilibrium structures	Experimental data
ABCHM[140]	Extended the potential by Ackland et al.[141] (a Finnis-Sinclair potential[111]) by adding a quadratic term in the embedding function to improve the melting point. Developed for crystallization kinetics from deeply undercooled melts	*Ab initio* and experimental data. Melting properties and crystalline properties
CuNi[136]	Defined by a Morse function[119], the universal equation of state[139] and a hyperbolic tangent which considers the shape of the 3d orbital. Developed for Cu-Ni alloys	Experimental data to train the Cu potential
EAM1[123]	Based on Morse potentials, unit step functions, and other terms. Widely used, especially for defect properties in copper.[142] Developed for energies and mechanical stability of non-equilibrium structures	Experimental data with *ab-initio* data for relative energies of hcp, bcc, and fcc phases, and energies of the fcc phase and a diatomic molecule under strong compression
Cu1[140]	Developed for crystallization kinetics from deeply undercooled melts.[142]	*Ab initio* and experimental data. Crystalline, liquid and melting point data
Cuu6[143]	Developed and defined in the same way as Cuu3 but used more accurate vacancy formation energies	Experimental data
Cuu3[144]	Defined by the universal equation of state[139] and the spherically averaged free-atom densities calculated from Hartree-Fock theory.[145-146] Developed for several fcc metals and alloys	Experimental data
Cu2[142]	Developed in the same way as Cu1, but added twin boundary energy to training data	*Ab initio* and experimental data. Melting properties, crystalline properties, twin boundary energy
MCu31[147]	The paper developed several EAM potentials to study the effects of stacking fault energy on dislocation nucleation[(a)]	*Ab initio* and experimental data. Melting properties, crystalline properties, twin boundary energy and stacking fault energy

Notes: (a) MCu31 was chosen for comparison here because it calculates stacking fault energies more accurately than the others.[37] The equation for the model itself was not provided, but as it was described as being based on Cu2 and fit to additional training data, we have assigned it the same complexity as Cu2. The acronyms of the models are either taken from the original paper or adapted from the Interatomic Potentials Repository.[37]

2.3.5 Enabling GP1, GP2, and GP3 in LAMMPS

GP1 can be used under the "pair_style eam/alloy" in LAMMPS. Cu_GP1.eam is the corresponding potential file. The following is an example input specification:

- pair_style eam/alloy
- pair_coeff * * Cu_GP1.eam Cu

GP2 and GP3 use "pair_style poet" and require the potential file Cu_GP2.poet or Cu_GP3.poet, respectively. The following is an example input specification:

- pair_style poet
- pair_coeff * * Cu_GP2.poet Cu

The poet pair_style can be compiled in LAMMPS following these steps:

1. Copy the files "pair_poet.cpp" and "pair_poet.h" (available upon request) to <lammps_main_directory>/src/MANYBODY and edit the file <lammps_main_directory>/src/Makefile.list by adding "pair_poet.cpp" and "pair_poet.h" to the end of the respective lines
2. *make* LAMMPS by including the "yes-manybody" flag

2.4 Results

2.4.1 Validating the machine learning algorithm

As a way of validating our approach, we tested the ability of our algorithm to rediscover the exact form of two interatomic potentials: the Lennard-Jones potential and the Sutton-Chen (SC) EAM potential. Specifically, we generated training datapoints with the Lennard-Jones or SC EAM interatomic potential models, and our genetic programming algorithm

identified the exact functional form of the interatomic potential model used for generating the training data. The training data for the Lennard-Jones potential were generated by taking 75 snapshots from molecular dynamics simulations. One snapshot was taken every 5000 steps with a time step of 1 fs using a 32-atom supercell. The 75 snapshots were obtained from molecular dynamics simulations in the following settings:

- 15 snapshots were obtained from an NVT ensemble at 80 K
- 15 snapshots from an NPT ensemble at 80 K and 100 kPa
- 15 snapshots from an NVT ensemble at 100 K
- 15 snapshots from an NPT ensemble at 100 K and 100 kPa
- 15 snapshots from an NVT ensemble at 20,000 K

In total, the training set consisted of 75 energies and 7200 components of force [101], and it was generated using the following parameterized Lennard-Jones model for argon[148]:

$$V_{LJ} = \sum_i \sum_j \left(\frac{49304.15}{r^{12}} - \frac{34.88}{r^6} \right) \tag{2.14}$$

where V_{LJ} is the potential energy of the system, the index i represents an atom in the structure, j is its neighbor and r is the distance between the two atoms. The goal was to identify equation (2.14), and the genetic programming algorithm found:

$$V = \sum_i \left(-50.18(983.04) \left(\sum_j (3.35r)^{-6.00} - \sum_j r^{-12.00} \right) \right) \tag{2.15}$$

which simplifies to the form of the Lennard-Jones potential in equation (2.14).

The next step for validating the genetic programming algorithm was to re-discover the SC EAM interatomic potential. We created the training data from 100 snapshots of 32-atom

molecular dynamics simulations, taking 1 snapshot every 100 steps with a time step of 1 fs. The snapshots were taken from molecular dynamics simulations in the NVT ensemble in the following way:

- 25 snapshots at 300 K
- 25 snapshots at 1600 K
- 25 snapshots at 3800 K
- 25 snapshots at 20,000 K

The training set consisted of 100 energies and 9600 components of forces, and it was generated using the following SC EAM interatomic potential model parametrized for copper:

$$V_{SC} = \sum_i \left(\sum_j \frac{644.52}{r^9} - \left(\sum_j \frac{527.62}{r^6} \right)^{0.5} \right) \tag{2.16}$$

The artificial intelligence algorithm found:

$$V = \sum_i \left(-0.73 - 2.53 \left(\left(-0.66(384.39) \sum_j r^{-9.00} \right) + \left(0.25 / \left(20.63 \sum_j r^{-6.00} \right) \right)^{-0.50} \right) \right)$$

(2.17)

Which gives the same form as the SC EAM in equation (2.16) after simplifying it, with an additional constant shift and a slight difference between the constant parameters that could be eliminated by tightening the convergence criterion for parameter optimization. The values of the parameters in the exponents were found to the second decimal place. This

was the first time that symbolic regression has been used to successfully rediscover a many-body interatomic potential.

2.4.2 Testing the symbolic regression algorithm: discovering new models for copper from *ab initio* data

Discovering novel interatomic potential models from quantum mechanical data is a more challenging task than identifying the Lennard-Jones and SC EAM models from data generated using the same models. This an essential metric of success for our genetic programming algorithm. Therefore, after re-discovering the exact form of a simple pair potential (i.e., Lennard-Jones model) and a many-body potential (i.e., the SC EAM model) with our genetic programming algorithm, we evaluated its ability to find potential models from data generated using an density functional theory [149] (DFT). Namely, we generated the data for this test by taking snapshots from molecular dynamics simulations using DFT, which belongs to the category of *ab initio* molecular dynamics (AIMD). We generated 150 snapshots from DFT molecular dynamics simulations on 32-atom supercells on fcc copper. We took 1 snapshot every 100 steps with a time step of 1 fs. We generated the data from the snapshots in the following way:

- 50 snapshots at 300K in the NVT ensemble
- 50 snapshots at 1400 K NVT ensemble
- 50 snapshots at 1400 K in the NPT at a pressure of 100 kPa

For the simulations at the high temperature of 1400 K, the copper structures did not maintain their fcc configuration. We collected 150 snapshots, which means that we generated datapoints corresponding to 150 energies, 14400 components of forces and 900

components of virial stress tensors[150]. We randomly split the data into training and validation, taking 75 snapshots for training and 75 snapshots for validation.

To implement the multi-objective optimization portion of the algorithm, we defined three metrics:

1. complexity, defined as the number of nodes on the tree representation of the interatomic potential model.
2. computational cost, defined as the number of summations over neighbors, as these typically consume most of the execution time.
3. fitness, defined as a weighted sum of the mean squared errors of the normalized energies, forces and stresses. The fitness is shown in equation (2.18) below. The data were normalized to unitless values as described in the methods section.

$$fitness = 1000 * (0.5MSE_{energy} + 0.4MSE_{force} + 0.1MSE_{stress}) \tag{2.18}$$

Then, we identified promising models by constructing a three-dimensional convex hull based on fitness, computational cost, and complexity. Some of the models on this hull are shown in Table 2.

Table 2. The 3-dimensional convex hull of models found by the machine learning algorithm.

Fitness	Cost*	Complexity	Expression
5393157	1	2	$\sum rf(r)$
1800.1	1	4	$\sum r^{-3.20} f(r)$
105.30	1	8	$\sum (649.17 r^{-9.83} - 0.09) f(r)$
54.144	1	10	$\sum (r^{10.20-5.49r} - 0.07) f(r)$
26.906	2	13	$\sum r^{10.20-5.49r} f(r) + 33.77 \left(\sum f(r) \right)^{-1}$
8.1584	2	15	$\sum r^{10.21-5.48r} f(r) + 1.19 \left(\sum 0.33^r f(r) \right)^{-1}$
7.8230	2	21	$\sum (r^{10.21-5.47r} - 0.21^r) f(r) + 0.97 \left(\sum 0.33^r f(r) \right)^{-1}$
7.8229	2	25	$0.999 \sum (r^{10.21-5.46r} - 0.21^r) f(r) + 0.97 \left(\sum 0.33^r f(r) \right)^{-1} + 5.76$
7.4131	4	19	$\sum r^{10.21-5.48r} f(r) + \left(3.07 \sum f(r) \right) \left(\sum 0.31^r f(r) \right)^{-1} \left(\sum rf(r) \right)^{-1}$
4.7294	3	28	$7.33 \sum r^{3.98-3.94r} f(r) + \left(27.32 - \sum (11.13 + 0.03 r^{11.74-2.93r}) f(r) \right) \left(\sum f(r) \right)^{-1}$
4.2932	4	29	$6.76 \sum r^{4.00-3.88r} f(r) + 17.25 \left(\sum f(r) \right) \left(\sum r^{11.68-3.07r} f(r) \right)^{-1} + 25.30 \left(\sum f(r) \right)^{-1}$

Notes: the models with fitness 7.8230 and 4.7294 are named GP1 and GP2 respectively. "Cost" is based on the number of summations. $f(r)$ is the smoothing function defined in Equation (2.12).

The genetic programming algorithm found models with a wide range of fitness values. The models on the range of worse fitness values (large fitness values) are pair potentials, and most of the best models (small fitness values) have forms that resemble the embedded atom model, or glue type potentials. In other words, the algorithm discovered EAM-type models from *ab initio* data. The EAM-type models are formed by of a sum of a pairwise term with

a repulsive component and a many-body (i.e., glue) type attractive term which consists of a nonlinear transformation of a sum over neighbors. The sum over neighbors represents the electron density, and the nonlinear transformation is the embedding function. We selected two of the models shown in Table 2, which we label GP1 and GP2, for further analysis because of their favorable tradeoff between simplicity and their prediction errors for the elastic constants.

Table 3. Errors on different properties for models on the 3-dimensional convex hull. The models are listed the order they appear in the table. C_{ij} are elastic constants, a_0 is the lattice parameter, $\Delta E_{bcc\text{-}fcc}$ is the energy difference between bcc and fcc phases, E_v is the fcc bulk vacancy formation energy, $E_{v,\ unrelaxed,\ 2\times2\times2}$ is the unrelaxed vacancy formation energy computed on a 2×2×2 supercell, E_m is the migration energy for fcc bulk vacancy diffusion, E_a is the activation energy for fcc bulk vacancy diffusion, $E_{dumbbell}$ is the dumbbell <100> formation energy, ν is the phonon frequency, and γ_{ISF} and γ_{USF} are the intrinsic and unstable stacking fault energies, respectively. $\bar{\gamma}$ is the average surface energy weighted according to the Wulff construction and γ_{abs} is the mean absolute surface energy over 13 surfaces.

Property	Metric	M1	M2	M3	M4	M5	M6	GP1	M8	M9	GP2	M11
Complexity[a]	Number of nodes	2	4	8	10	13	15	21	25	22	28	23
C_{11}[a]	% error	4081.6	-85.7	5.1	24.6	40.2	9.1	5.8	5.8	28.2	-0.7	2.3
C_{12}[a]	% error	3827.2	-84.5	-35.7	-17.4	21.1	12.9	7.0	7.0	24.8	0.5	1.6
C_{44}[a]	% error	4872.1	-91.7	3.3	32.5	10.3	-5.5	-2.0	-1.9	7.1	-1.2	-3.6
a_0 (fcc)	% error	-	95.5	-0.7	-1.4	-0.5	-0.2	-0.3	-0.3	-0.2	0.3	0.2
a_0 (bcc)	% error	23.0	100.8	7.5	-2.4	-1.2	0.2	0.1	0.0	-0.6	-0.1	-0.2
$\Delta E_{bcc\text{-}fcc}$	pred. – ref. (meV/at.)	43026	-36	74	97	89	8	8	8	42	4	12
$\Delta E_{hcp\text{-}fcc}$	pred. – ref. (meV/at.)	-5	-5	18	13	12	-3	-3	-3	3	-2	-2
$E_{v,\ unrelaxed,\ 2\times2\times2}$	pred. – ref. (meV)	-1117	-1117	167	224	66	41	32	33	-71	-123	-207
E_v	pred. – ref. (meV)	-970	-970	104	249	76	138	138	33	21	2	-89
E_m	pred. – ref. (meV)	10054	-709	-286	-79	-157	-133	-106	139	-72	-37	42
E_a	pred. – ref. (meV)	9084	-1679	-182	170	-81	6	32	-106	-51	-34	-47
$E_{dumbbell}$	pred. – ref. (meV)	-6×10^{24}	-2925	99	737	372	-28	49	33	220	-56	-41
$v_L(X)$	% error	206.3	-89.5	10.5	21.1	12.8	7.0	8.2	8.3	7.4	3.2	1.9
$v_T(X)$	% error	205.9	-89.5	-5.4	9.8	1.5	-0.9	0.7	0.9	2.0	0.0	-0.8
$v_L(L)$	% error	200.9	-89.7	13.2	22.1	13.6	5.4	6.5	6.5	6.2	0.5	-0.7
$v_T(L)$	% error	223.6	-88.9	0.8	15.3	6.5	-3.9	-2.2	-2.0	2.6	-3.0	-2.0
$v_L(K)$	% error	209.9	-89.4	9.4	20.7	12.5	7.8	9.1	9.1	8.0	4.3	2.9
$v_{T1}(K)$	% error	214.5	-89.2	-4.1	11.3	3.2	0.0	1.6	1.8	3.0	0.1	0.1
$v_{T2}(K)$	% error	211.0	-89.4	8.7	20.0	11.7	6.2	6.9	7.6	6.8	2.2	1.4
γ_{ISF}	pred. – ref. (mJ/m^2)	-90	-46	188	158	138	-27	-29	-29	44	-20	-19
γ_{USF}	pred. – ref. (mJ/m^2)	-216	-172	42	88	30	-49	-44	-44	-2	-31	-29
$\bar{\gamma}$	% error	-100.0	-100.0	-1.4	11.3	41.6	-3.0	-7.6	-7.6	2.1	-12.6	-19.1
γ_{abs}	| % error |	100.0	100.0	7.3	7.9	37.3	3.8	2.3	2.3	3.9	7.2	16.2
$\gamma_{(100)}$	% error	-100.0	-100.0	-2.1	0.9	26.8	-3.7	-8.2	-8.2	-1.7	-14.2	-21.7
$\gamma_{(110)}$	% error	-100.0	-100.0	-7.3	6.2	29.6	3.9	-1.3	-1.2	2.1	-9.7	-17.4
$\gamma_{(111)}$	% error	-100.0	-100.0	20.2	37.7	50.2	0.0	-4.8	-4.7	4.4	-10.5	-17.0

Notes: (a) This property was used for model selection from the convex hull.

The models that we identified with our approach, GP1 and GP2, resemble known potential models in the general EAM framework (Table 4); see equation (2.19) below for the EAM model or equation (2.9) from the Background chapter for more details.

However, there are some notable differences. They have much simpler functional forms than most other copper potential models (Table 4), and they have a different functional form for the embedding term. It is common in EAM-type potential models for the embedding function to be the negative square root of the density; this can be derived from the second moment approximation.[113] In GP1 and GP2, the attractive term instead depends on the positive inverse of a sum over pairwise interactions. This embedding function is bounded in the limit of high densities and diverges to infinity in the limit of zero density, which is different than the other models. For example, in GP1, the simpler of the two models, the embedding function is the inverse of the density. In GP2, the embedding function also has the inverse of the density, but it is multiplied by a sum of pairwise interactions to form the glue term. Interestingly, GP1 and GP2 terms of the form r^{a-br} that grow following a power law before decaying superexponentially.

$$E_{tot} = \frac{1}{2}\sum_{ij} V(r) + \sum_{i} F(\bar{\rho}_i) \tag{2.19}$$

Table 4. EAM-type interatomic potentials for Cu near the Pareto frontier of maximum absolute percent error on elastic constants and complexity. The Pareto frontier can be defined as follows: no model is both less complex and has less error than a model in the Pareto frontier.

Name	Expression
SC[138]	$E_i = \sum_j \frac{644.52}{r^9} f(r) - \left(\sum_j \frac{527.62}{r^6} f(r) \right)^{0.5}$
GP1	$E_i = \sum_j (r^{10.21-5.47r} - 0.21^r) f(r) + 0.97 \left(\sum_j 0.33^r f(r) \right)^{-1}$
GP2	$E_i = 7.33 \sum_j r^{3.98-3.94r} f(r) + \left(27.32 - \sum_j \left(11.13 + 0.03 r^{11.74-2.93r} \right) f(r) \right) \left(\sum_j f(r) \right)^{-1}$
EAM2[123]	$E_i = \sum_j E_1 \left(e^{-2\alpha(r-r_0)} - 2e^{-\alpha(r-r_0)} \right) f(r) + F \left(\sum_j r^6 (e^{-\beta r} + 2^9 e^{-2\beta r}) f(r) \right)$ $\sum_i F(\bar{\rho}_i) = E(L) - \frac{1}{2} \sum_i \sum_j E_1 \left(e^{-2\alpha(r-r_0)} - 2e^{-\alpha(r-r_0)} \right) f(r)$ $E(L) = -E_{sub}(1+a^*)e^{-a^*}$ $a^* = (a/a_0 - 1)/(E_{sub}/9B\Omega)^{1/2}$
ABCHM[140]	$E_i = \sum_j \varphi(r) f(r) + 1.57 \cdot 10^{-5} \left(\sum_j \psi(r) f(r) \right)^2 - \left(\sum_j \psi(r) f(r) \right)^{0.5}$ $\varphi(r) = \begin{cases} e^{0.82+16.01r-15.73r^2+3.80r^3}, & 1 < r < 1.9 \\ +0.62(4.43-r)^3, & 1.9 < r < 4.43 \\ -3.02(4.17-r)^3, & 1.9 < r < 4.17 \\ +2.84(4.04-r)^3, & 1.9 < r < 4.04 \\ -0.41(3.62-r)^3, & 1.9 < r < 3.62 \\ +0.65(3.13-r)^3, & 1.9 < r < 3.13 \\ +0.81(2.56-r)^3, & 1.9 < r < 2.56 \end{cases}$ $\psi(r) = \begin{cases} 0.21(4.43-r)^3, & 1.9 < r < 4.43 \\ +0.36(3.62-r)^3 & 1.9 < r < 3.62 \end{cases}$

CuNi[136]	$E_i = \frac{1}{2}\sum_j \left(D_M \left[1 - e^{-\alpha_M (r - R_M)} \right]^2 - D_M \right) f(r) + F(\bar{\rho}_i)$ $\bar{\rho}_i = \sum_j \tanh\left(20 r^2\right) \left\{ r^6 \left(e^{-\beta r} + 2^9 e^{-2\beta r} \right) + \frac{\sigma^{(1)}}{\mu^{(1)}} e^{-\frac{1}{2}\left[\mu^{(1)}(r - R_B)\right]^2} - 0.1\sigma^{(1)} e^{-\frac{1}{2}\left[\mu^{(1)}(r - (R_B + 0.5))\right]^2} \right\} f(r)$ $\sum_i F(\bar{\rho}_i) = E(L) - \frac{1}{2}\sum_i \sum_j \left(D_M \left[1 - e^{-\alpha_M (r - R_M)} \right]^2 - D_M \right) f(r)$ $E(L) = -E_{sub}(1 + a^*) e^{-a^*}$ $a^* = (a / a_0 - 1) / (E_{sub} / 9B\Omega)^{1/2}$
EAM1[123]	$E_i = \sum_j \begin{pmatrix} \left[\begin{array}{l} E_1 \left(e^{-2\alpha_1 (r - r_0^{(1)})} - 2e^{-\alpha_1 (r - r_0^{(1)})} \right) + \\ E_2 \left(e^{-2\alpha_2 (r - r_0^{(2)})} - 2e^{-\alpha_2 (r - r_0^{(2)})} \right) + \delta \end{array} \right] f(r) \\ -\sum_{n=1}^{3} \left(H\left(r_s^{(n)} - r\right) S_n (r_s^{(n)} - r)^4 \right) \end{pmatrix} + F(\bar{\rho}_i)$ $if\,(\bar{\rho}_i < 1) : F(\bar{\rho}_i) = F^{(0)} + 0.5F^{(2)} (\bar{\rho}_i - 1)^2 + \sum_{n=1}^{4} \left(q_n (\bar{\rho}_i - 1)^{n+2} \right)$ $else : F(\bar{\rho}_i) = \frac{F^{(0)} + 0.5F^{(2)} (\bar{\rho}_i - 1)^2 + q_1 (\bar{\rho}_i - 1)^3 + Q_1 (\bar{\rho}_i - 1)^4}{1 + Q_2 (\bar{\rho}_i - 1)^3}$ $where : \bar{\rho}_i = \sum_j \left(\left[a e^{-\beta_1 (r - r_0^{(3)})^2} + e^{-\beta_2 (r - r_0^{(4)})} \right] f(r) \right)$

Note: All potentials are in units of eV and Å. $f(r)$ is a smoothing function; for GP1 and GP1 it is defined in Equation (2.12). EAM2 and CuNi defined the embedding function to match a universal equation of state[139].

An important feature of interatomic potential models is how well they predict properties that were not included in the training set, particularly properties derived from structures that have atomic environments different to the ones included in the training set. GP1 and GP2 did not include surfaces in their training data, and they largely avoid the severe underprediction of surface energies that are common for embedded-atom type models (Figure 9 and Table 12).[123] GP1 and GP2 also demonstrate high predictive power for condensed phases and defects that were not included in the training data.

2.4.3 Assessing the transferability of the interatomic potential models

GP1 and GP1 have mean absolute errors on the training energies, the components of force, and the components of the virial stress tensor that are similar to the mean absolute errors on the validation energies, forces and stresses (Figure 4). This suggests that neither GP1 nor GP2 overfit their training data, which is expected due to their simplicity. A more complex model may overfit the data given the relatively small number of training points. The previous validation metric measures the performance of GP1 and GP2 on atomic environments like the ones used in training. A similar validation metric is how well GP1 and GP2 reproduce the radial distribution function of the liquid state, since snapshots of molecular dynamics simulations of the liquid phase were included in their training data GP1 and GP2 do a good job at reproducing the radial distribution function of the liquid phase (Figure 5).

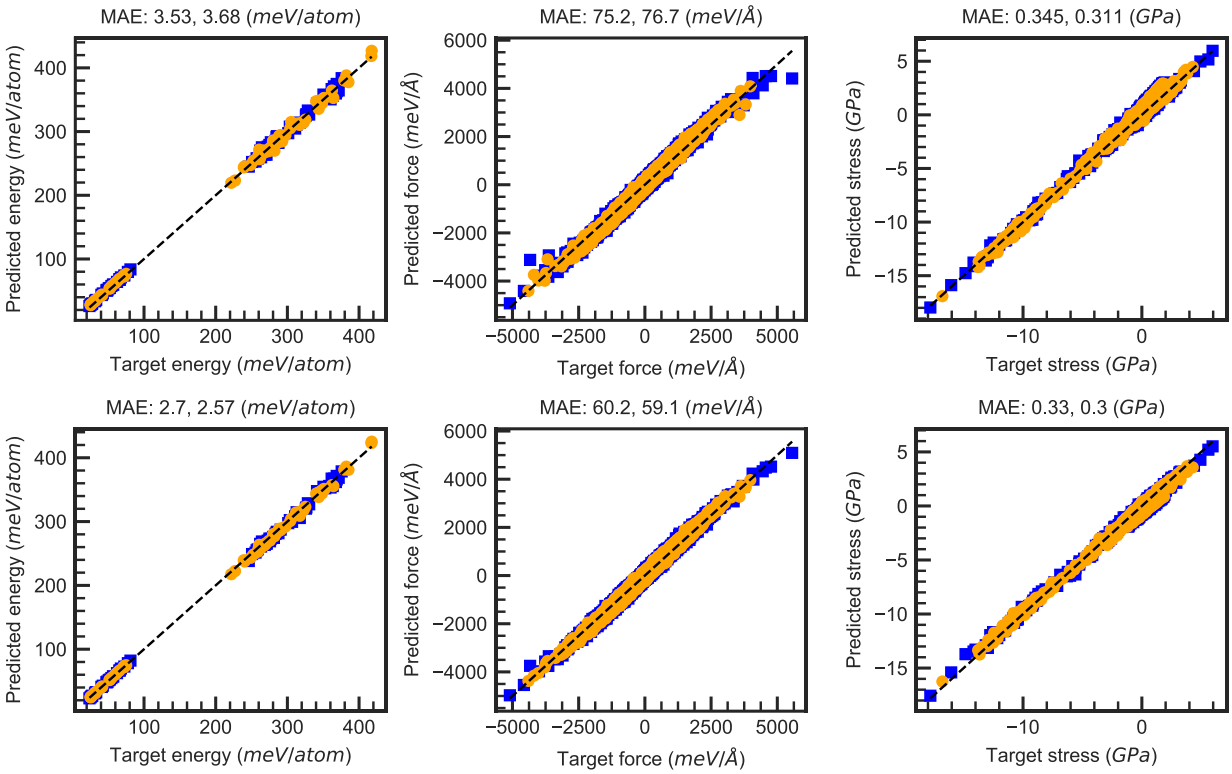


Figure 4. Parity plots of training (orange) and validation (blue) energies, components of force and components of the virial stress tensor for the interatomic potential GP1 (a) and GP2 (b). The black dashed line is the identity. The mean absolute error (MAE) is presented above each sub-figure for validation and training data respectively.

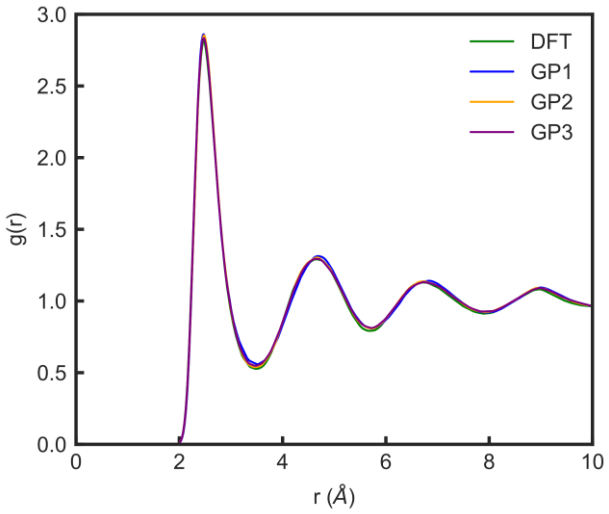


Figure 5. Radial distribution functions of liquid copper at 1400K

We also compared GP1 and GP2 against other EAM-type interatomic potential models from the literature. Three widely used properties used as a benchmark of the performance of copper potential models are the constants C11, C22, and C44 of the fcc structure. With the elastic constants, we can compare different copper potential models for which elastic constant data is available (Table 5).

Table 5. Errors on elastic constants and lattice parameters of EAM-type interatomic potentials for Cu

Model	Complexity	C11	C12	C44	a0 (FCC)	a0 (BCC)
Description	Number of nodes	Error	Error	Error	Error	Error
Units	Count	%	%	%	%	%
SC	15	-3.6[a, b]	3.8[a, b]	-29.4[a, b]	0.0[a]	-
GP1	21	5.8[c]	7.0[c]	-2.0[c]	-0.3[c]	0.1[c]
GP3	26	2.9[c]	2.5[c]	-0.4[c]	0.2[c]	-0.2[c]
GP2	28	-0.7[c]	0.5[c]	-1.2[c]	0.3[c]	-0.1[c]
EAM2	113	1.9[a]	0.2[a]	0.0[a]	0.0[a]	-
ABCHM	146	-0.6[a]	-4.1[a]	-6.6[a]	-0.7[c]	2.4[c]
CuNi	150	0.1[a]	0.0[a]	0.0[a]	0.0[a]	0.9[c]
EAM1	158	-0.1[a]	0.1[a]	0.5[a]	0.0[a]	-
Cu1	348	2.9[a]	4.1[a]	10.5[a]	0.0[c]	0.0[c]
Cuu6	503	-1.2[a]	1.2[a]	4.2[a]	0.0[a]	-
Cuu3	503	-1.8[a]	1.2[a]	0.3[a]	0.0[a]	-
Cu2	584	2.4[a]	3.3[a]	10.5[a]	0.0[c]	0.0[c]
MCu31	584	-1.2[a]	6.5[a]	13.2[a]	0.0[c]	-

Notes: properties in orange were used for training and properties in blue were used for validation. (a) experiment target data. (b) fit to bulk modulus. (c) *ab initio* calculation target data.

We are interested in developing simple and accurate interatomic potential models. The Pareto frontier is a good way of identifying optimal models considering the tradeoff between complexity and accuracy of the models. The Pareto frontier can be defined in the following way: no model is both more accurate and less complex than a model in the Pareto set. For evaluating our models in terms of complexity and simplicity, we plotted the maximum percent error in predicted elastic constants against the complexity of the model, as measured by number of nodes (Figure 6). The Pareto frontier significantly improves with the interatomic potentials discovered by the machine learning algorithm presented in this thesis. The errors of GP1 and GP2 are comparable to the most accurate EAM-type interatomic potential models for Cu, and their complexity is comparable to the simplest.

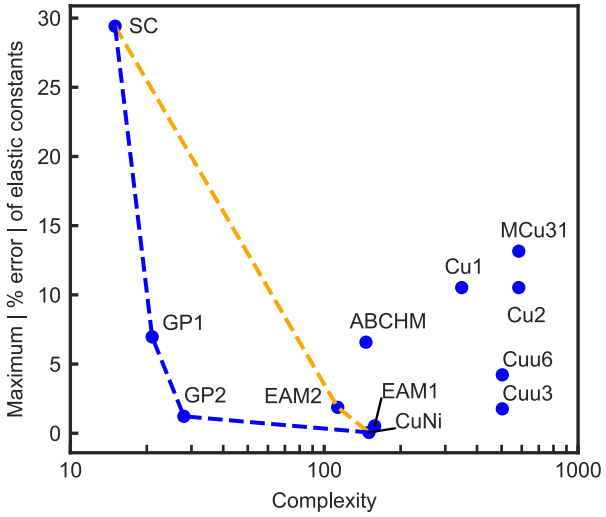


Figure 6. Pareto frontiers of EAM-type interatomic potentials for copper. No model has less error and is less complex than a model in the Pareto frontier. The orange dashed line was the Pareto frontier before the development of GP1 and GP2, and the blue dashed line is the new Pareto frontier. The percent error for each model was evaluated against the model's own target values, described in the Methods section of this chapter. Complexity was measured by the number of nodes in the tree representation of the model. Because the smoothing function for some models is unknown, to construct this plot each smoothing function was counted as 2 nodes, representing the smoothing function and a multiplication operation. Sources: SC[138], ABCHM[140], Cu1[140], EAM1[123], EAM2[123], Cu2[142], Cuu6[143], Cuu3[144] and CuNi[136]. The interatomic potentials were found in the Interatomic Potentials Repository.[37] GP1 and GP2 were developed as part of this thesis.

Using the average of the absolute percent errors instead of the maximum of the absolute percent errors on elastic constants (Figure 7) does not change the insights obtained from Figure 6.

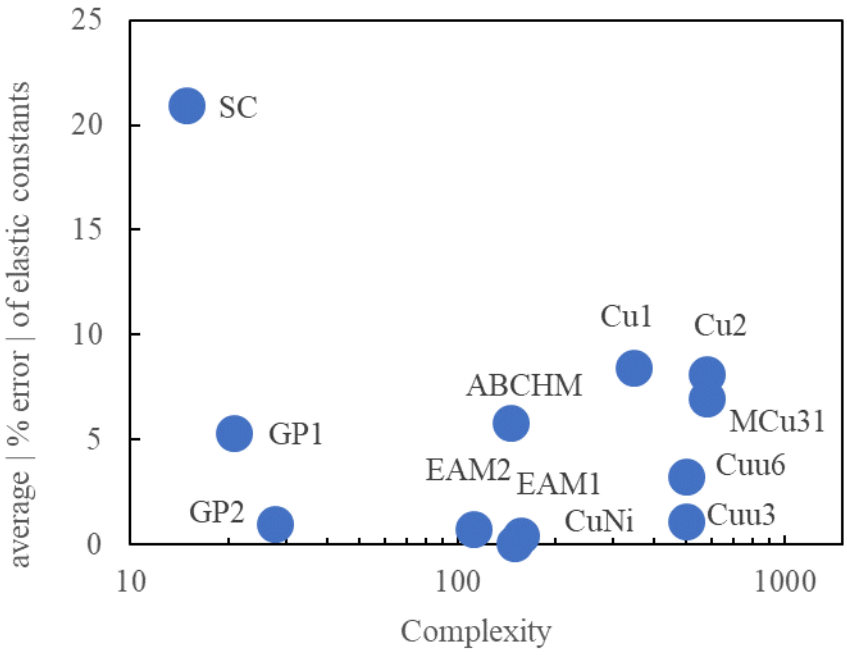


Figure 7. Average (instead of maximum) absolute error on elastic constants. The Pareto frontier does not change from the one calculated using maximum error.

Other properties calculated with GP1 and GP2 show good agreement with the values computed with DFT (Table 6), even though these properties were not included in the training dataset of GP1 and GP2. In contrast, many of the properties listed in Table 6 were included in the training set of the literature models near the Pareto frontier (Figure 6). As expected, the interatomic potential models from the literature that included the properties in their training set also show good agreement with their target values. However, the literature models are more complex, except for SC. For example, the errors on the elastic constants predicted by GP2 are almost as small as for EAM1, and the simpler model GP1 has errors on elastic constants that are comparable to ABCHM (Table 6).

Table 6. Error of the values predicted by EAM-type interatomic potentials for copper relative to the respective reference. The models displayed in this table are near the Pareto frontiers in Figure 6, values of other potentials are in Tables S2 to S7. C_{ij} are elastic constants, a_0 is the lattice parameter, ΔE (bcc-fcc) is the energy difference between bcc and fcc phases, E_v is the fcc bulk vacancy formation energy, $E_{v\ (unrelaxed,\ 2\times2\times2)}$ is the unrelaxed vacancy formation energy computed on a 2×2×2 supercell, E_m is the migration energy for fcc bulk vacancy diffusion, E_a is the activation energy for fcc bulk vacancy diffusion, $E_{dumbbell}$ is the dumbbell <100> formation energy, ν is the phonon frequency, and γ_{ISF} and γ_{USF} are the intrinsic and unstable stacking fault energies, respectively.

Property	Metric	SC	GP1	GP3	GP2	EAM2	ABCHM	CuNi	EAM1
Complexity	Number of nodes	15	21	26	28	113	146	150	158
C_{11}	% error	-3.6[a]	5.8[b]	2.9[b]	-0.7[b]	1.9	-0.6	0.1	-0.1
C_{12}	% error	3.8[a]	7.0[b]	2.5[b]	0.5[b]	0.2	-4.1	0.0	0.1
C_{44}	% error	-29.4[a]	-2.0[b]	-0.4[b]	-1.2[b]	0.0	-6.6	0.0	0.5
a_0 (fcc)	% error	0.0	-0.3	0.2	0.3	0.0	-0.7	0.0	0.0
a_0 (bcc)	% error	-	0.1	-0.2	-0.1	-	2.4	0.9	-
ΔE (bcc – fcc)	pred. – ref. (meV/atom)	-	8	12	4	2[c]	-11	-13	2
ΔE (hcp – fcc)	pred. – ref. (meV/atom)	-	-3	-1	-2	-6[c]	-2	-4	-4
$E_{v\ (unrelaxed,\ 2\times2\times2)}$	pred. – ref. (meV)	-	32	-106	-123	-	80	-	-
E_v	pred. – ref. (meV)	-	138	12	2	-17	-	6	-3
E_m	pred. – ref. (meV)	-	-106	-49	-37	-20	-	-20	-21
$E_a = E_v + E_m$	pred. – ref. (meV)	-	32	-36	-34	-37	-24[d]	-14[d]	-24
$E_{dumbbell}$	pred. – ref. (meV)	-	49	-15	-56	-271	250	452	-437
$\nu_L(X)$	% error	-	8.2	4.1	3.2	7.6	-	0.0	6.0
$\nu_T(X)$	% error	-	0.7	0.1	0.0	1.2	-	-0.2	0.8
$\nu_L(L)$	% error	-	6.5	1.7	0.5	6.6	-	-0.8	4.6
$\nu_T(L)$	% error	-	-2.2	-2.3	-3.0	-1.5	-	-9.4	-2.6
$\nu_L(K)$	% error	-	9.1	5.1	4.3	8.0	-	-0.8	5.4
$\nu_{T1}(K)$	% error	-	1.6	0.3	0.1	2.4	-	1.1	1.1
$\nu_{T2}(K)$	% error	-	6.9	3.0	2.2	9.0	-	0.9	7.0
γ_{ISF}	pred. – ref. (mJ/m^2)	-	-29	-6	-20	-9	-	0	-1
γ_{USF}	pred. – ref. (mJ/m^2)	-	-44	-27	-31	-	-	-	-

Note: properties in orange font were used for training and properties in blue font were not used for training. Properties for which target values are not available are marked with a "-". (a) SC was fit to the bulk modulus. (b) elastic constants were used to select GP1, GP2 and GP3 from the convex hull. (c) fit to ensure that $E_{fcc} < E_{bcc}$ and $E_{fcc} < E_{hcp}$. (d) fit to vacancy formation energy.

The GP1 and GP2 models perform well on properties involving hcp and bcc phases, even though no hcp or bcc data were included in the training set. For the bcc lattice constant, the relative energy between the fcc and bcc phases, and the relative energy between fcc and hcp phases, GP1 and GP2 perform comparably to models that were trained on those data points and outperform most models that were not trained on them (Table 5, Table 6, and Table 7).

Table 7. Errors on difference between energies of FCC, BCC and HCP phases of EAM-type interatomic potentials for Cu

Model	Complexity	ΔE(BCC-FCC)	ΔE(HCP-FCC)
Description	Number of nodes	Pred.-Targ.	Pred.-Targ.
Units	Count	meV/atom	meV/atom
GP1	21	8	-3
GP3	26	12	-1
GP2	28	4	-2
EAM2	113	2[a]	-6[a]
ABCHM	146	-11	-2
CuNi	150	-13	-4
EAM1	158	2	-4
Cu1	348	5	0
Cu2	584	6	1
MCu31	584	0	-6

Notes: properties in orange were used for training and properties in blue were used for validation. All values are from *ab initio* calculations. (a) EAM2 was fit subject to the requirement that the fcc structure be more stable than bcc or hcp.

GP2 has a very good accuracy on the dilute vacancy formation energy, with an error of 2 meV relative to DFT; see the Methods section for details about the dilute vacancy formation energy. GP1 performs less well, with an error of 138 meV. It is difficult to compare the errors of the vacancy formation energy of GP1 and GP2 against EAM-type models from the literature because literature models included the vacancy formation energy in their training set. However, we can compare our models against a neural network

potential [151] that reports validation errors on vacancy formation energies. The neural network potential has an error of 146 meV on the extrapolated vacancy formation energy, comparable to GP1 (Table 8). We have more discussion on the neural network potential later in this chapter.

Table 8. Comparison between genetic programming potentials and a neural network potential.[151]

Feature	Units or details	GP1	GP3	GP2	Neural network
Parameters	count	5	7	8	2521
Atomic environments in training set	count	2400	2651	2400	554,187
Energy MAE	meV/atom	3.5	2.5	2.7	2.2
Force MAE	meV/Å	75	62	60	56
C_{11}	% error	5.8	2.9	-0.7	2.3
C_{12}	% error	7.0	2.5	0.5	-3.3
C_{44}	% error	-2.0	-0.4	-1.2	3.8
a_0 (fcc)	% error	-0.3	0.2	0.3	0.0
a_0 (bcc)	% error	0.1	-0.2	-0.1	0.1
ΔE (bcc-fcc)	pred. - ref. (meV/atom)	8	12	4	-4
ΔE (hcp-fcc)	pred. - ref. (meV/atom)	-3	-1	-2	-7
E_v	pred. - ref. (meV)	138	12	2	146[a]
$E_{v\ (relaxed,\ 3\times3\times3)}$	pred. - ref. (meV)	111	-15	-26	106
$E_{v\ (relaxed,\ 2\times2\times2)}$	pred. - ref. (meV)	53	-75	-88	10
$E_{v\ (unrelaxed)}$	pred. - ref. (meV)	142	2	-15	147[a]
$E_{v\ (unrelaxed,\ 3\times3\times3)}$	pred. - ref. (meV)	109	-31	-47	103
$E_{v\ (unrelaxed,\ 2\times2\times2)}$	pred. - ref. (meV)	32	-106	-123	-1
(100) surface energy	% error	-8.2	-10.5	-14.2	0.5
(110) surface energy	% error	-1.3	-6.1	-9.7	1.5
(111) surface energy	% error	-4.8	-5.1	-10.5	-0.4

Notes: properties in orange were used for training and properties in blue were used for validation. All the target properties were computed by DFT. The DFT dilute vacancy formation energies were obtained by linearly extrapolating the values at 2×2×2 and 3×3×3 with respect to the inverse of the supercell size.[132] The dilute values for GP1 and GP2 were computed with a 6x6x6 supercell. (a) The dilute values for the neural network were computed by extrapolating the values at 2×2×2 and 3×3×3.

The vacancy activation energy, equation (2.20) can be more important than the vacancy formation energy alone for atomistic simulations of vacancy diffusion; see Vineyard's transition state theory model (equation (2.21)). The errors of GP1 and GP2 on the vacancy activation energy for vacancy-mediated diffusion are comparable to models that were trained on that value (Table 9). One reason for this low error is that the error in vacancy formation energy of GP1 is largely offset by an error in the opposite direction for the vacancy migration energy. The vacancy activation energy is:

$$E_a = E_v + E_m \tag{2.20}$$

where E_v is the vacancy formation energy, and E_m is the vacancy migration energy. The vacancy activation energy is needed for computing Vineyard's vacancy hop frequency in the following Arrhenius-like expression:

$$k = \nu^* e^{-\frac{E_a}{k_B T}} \tag{2.21}$$

where k is the rate constant or hop frequency of a vacancy, ν^* is the hop attempt frequency (or frequency of vibration), and the exponential term provides the probability of success.

Table 9. Errors on bulk vacancy formation energy, migration energy, activation energy and dumbbell <100> formation energy of EAM-type interatomic potentials for Cu

Model	Complexity	E_v (unrelaxed,2×2×2)	E_v	E_m	E_a	$E_{dumbbell}$
Description	Num. of nodes	Pred.-Targ.	Pred.-Targ.	Pred.-Targ.	Pred.-Targ.	Pred.-Targ.
Units	Count	meV	meV	meV	meV	meV
GP1	21	32[a]	138[a]	-106[a]	32[a]	49[a]
GP3	26	-106[a]	12[a]	-49[a]	-36[a]	-15[a]
GP2	28	-123[a]	2[a]	-37[a]	-34[a]	-56[a]
EAM2	113	-	-17[b]	-20[b]	-37[b]	-271[b]
ABCHM	146	80[a]	-	-	-24[b, c]	250[a]
CuNi	150	-	6[b]	-20[b]	-14[b, c]	452[a]
EAM1	158	-	-3[b]	-21[b]	-24[b]	-437[b]
Cu1	348	-9[a]	-	-	-2[b, c]	-120[a]
Cuu6	503	-	30[b]	-20[b]	-50[b, c]	-
Cuu3	503	-	-20[b]	-40[b]	-120[b, c]	-
Cu2	584	53[a]	-	-	-26[b, c]	-120[a]
MCu31	584	-	-	-	-	-110[a]

Notes: properties in orange were used for training and properties in blue were used for validation. (a) *ab initio* target data. (b) experimental target data. (c) fitted to the vacancy formation energy

GP1 and GP2 perform much better than the other models for the formation energy of a dumbbell defect, which is a commonly used validation measure for interatomic potential models of copper (Table 9). None of the evaluated models included the dumbbell formation energy in their training data, and all but Sutton Chen reported comparisons with their target values, making this a useful assessment of the predictive power of the different models. The absolute errors for GP1 and GP2 are only 49 meV and 56 meV respectively, as compared to absolute prediction errors in the range of 250 to 452 meV for the other models in Table 6.

Phonon spectra are an important metric for assessing the quality of interatomic potential models. They consider the curvature (second derivatives) of the total energy of the system

around the stable state.[136, 152] GP1 and GP2 demonstrate good predictive accuracy on phonon frequencies, which were not included in their training set (Figure 8). The average absolute of GP2 is lower than the average absolute all other models on phonon frequencies used for validation (Table 10), with a mean absolute error of 2.0%. Even though EAM1 trained on $\nu_L(X)$ and $\nu_T(X)$, GP2 has a lower error on these phonon frequencies. GP1 does not do as well as GP2 on phonon frequencies, performing on average slightly better than EAM2 but worse than EAM1 and CuNi. The phonon dispersion curves clearly show a better performance for GP2 than for GP1 (Figure 8). The strong performance of GP2 on phonon frequencies and elastic constants suggests that it does well at capturing the curvature of local minima on the potential energy surface, but it may not do as well in states away from the local minima, such as the vacancy formation energy of an unrelaxed 2×2×2 fcc unit cell (Table 6).

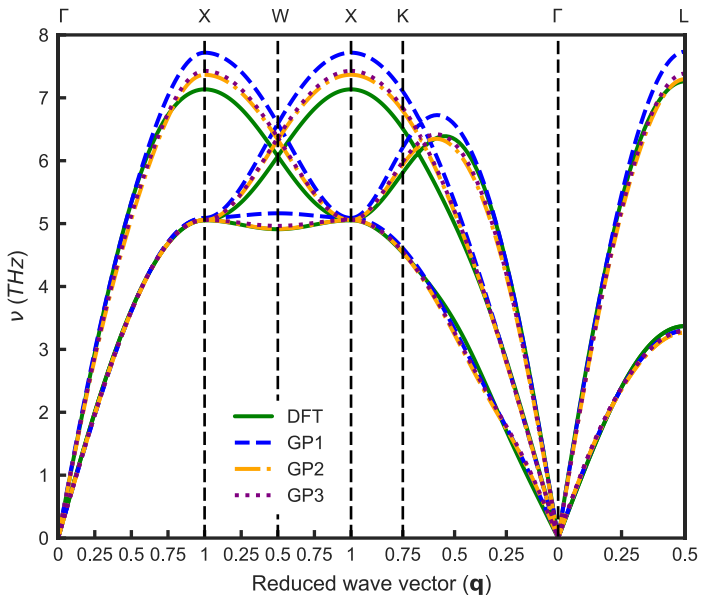


Figure 8. Calculated phonon dispersion curves for DFT, GP1, GP2, and GP3.

Table 10. Errors on phonon frequencies of EAM-type interatomic potentials for Cu

Model	Complexity	$\nu_L(X)$	$\nu_T(X)$	$\nu_L(L)$	$\nu_T(L)$	$\nu_L(K)$	$\nu_{T1}(K)$	$\nu_{T2}(K)$
Description	Number of nodes	Error	Error	Error	Error	Error	Error	Error
Units	Count	%	%	%	%	%	%	%
GP1	21	8.2[a]	0.7[a]	6.5[a]	-2.2[a]	9.1[a]	1.6[a]	6.9[a]
GP3	26	4.1[a]	0.1[a]	1.7[a]	-2.3[a]	5.1[a]	0.3[a]	3.0[a]
GP2	28	3.2[a]	0.0[a]	0.5[a]	-3.0[a]	4.3[a]	0.1[a]	2.2[a]
EAM2	113	7.6[b]	1.2[b]	6.6[b]	-1.5[b]	8.0[b]	2.4[b]	9.0[b]
CuNi	150	0.0[b]	-0.2[b]	-0.8[b]	-9.4[b]	-0.8[b]	1.1[b]	0.9[b]
EAM1	158	6.0[b]	0.8[b]	4.6[b]	-2.6[b]	5.4[b]	1.1[b]	7.0[b]

Notes: properties in orange were used for training and properties in blue were used for validation. (a) *ab initio* target data. (b) experimental target data.

GP1 and GP2 underestimate the formation energy of the stable intrinsic stacking fault (Table 11) to a greater extent than the other EAM-type models that report a comparison to

this value. The largest absolute error, 29 mJ / m^2 for GP1, is 10.2 meV / atom along the (111) plane of the fault. GP1 and GP2 similarly underestimate the formation energy of the unstable stacking fault, but it is hard to assess how this compares to other EAM-type models as none of the other models reports a benchmark value for the unstable stacking fault.

Table 11. Prediction errors for the intrinsic stacking fault (γ_{ISF}) energy and the unstable stacking fault (γ_{USF}) energy

Model	Complexity	pred. – ref. (mJ/m^2)	pred. – ref. (mJ/m^2)
Description	Number of nodes	γ_{ISF}	γ_{USF}
Units	Count	%	%
GP1	21	-29[a]	-44[a]
GP3	26	-6[a]	-27[a]
GP2	28	-20[a]	-31[a]
EAM2	113	-9[b]	-
CuNi	150	0[b]	-
EAM1	158	-1[b]	-
MCu31	584	6[b]	1[a]

Notes: properties in orange were used for training and properties in blue were used for validation. (a) *ab initio* target data. (b) experimental target data.

It is well known that the surface formation energies computed with EAM-type models tend to be lower than the surface energies from *ab initio* methods and experiments. The surface energies predicted by EAM-type models trained on *ab initio* calculations for copper are about 40-50% below their target values for the (100), (110) and (111) surfaces (Table 12). GP1 underpredicts these surface energies by only 8%, 1% and 5% respectively, and GP2 underpredicts them by 14%, 10% and 10% respectively (Figure 9). Evaluating the performance of GP1 and GP2 in calculating surface energies against interatomic potential models from the literature that used experimental data in the fitting process is more difficult because only the average of the experimental surface energies is available.[123, 136, 144] To

make this comparison, we have used GP1 and GP2 to calculate the weighted average surface energies over 13 different low-index surface facets, where the weights are based on the relative surface areas in Wulff constructions (details are provided in the Methods of this chapter)[134]. EAM1 and Cuu3 underpredict the weighted surface energies by about 30%, and CuNi overpredicts the weighted surface energies by about 10% (Table 12).[140] In comparison, GP1 underpredicts the weighted surface energies by 8% and GP2 by 13%. GP1-predicted surface energies are the most accurate of any of the evaluated EAM-type potential models relative to its target values.

Table 12. Prediction errors for surface energies of EAM-type interatomic potentials for Cu

Model	Complexity	%error of weighted surface energy	% error	% error	% error	Mean absolute % error.
Description	Number of nodes	13 surfaces	(100)	(110)	(111)	13 surfaces
Units	Count	%	%	%	%	%
GP1	21	-7.6[a]	-8.2[a]	-1.3[a]	-4.8[a]	2.3[a]
GP3	26	-7.4[a]	-10.5[a]	-6.1[a]	-5.1[a]	4.4[a]
GP2	28	-12.6[a]	-14.2[a]	-9.7[a]	-10.5[a]	7.2[a]
ABCHM	146	-	-49.7[a]	-47.4[a]	-53.8[a]	-
CuNi	150	9.8[b]	-	-	-	-
EAM1	158	-28.4[b]	-	-	-	-
Cu1	348	-	-50.0[a]	-48.4[a]	-53.8[a]	-
Cuu3	503	-31.8[b]	-	-	-	-
MCu31	584	-	-39.3[a]	-37.7[a]	-40.1[a]	-

Notes: properties in orange were used for training and properties in blue were used for validation. (a) *ab initio* target data. (b) experimental target data.

The performance of GP1 and GP2 on surface energies is remarkable because there were no surfaces in the training set; this is a case of machine-learning potential models demonstrating extrapolative predictive ability. Similarly, both GP1 and GP2 demonstrated

excellent predictive accuracy for the dumbbell defect compared to the other models, indicating that they are able to accurately predict energies in both low-coordination and high-coordination environments. There are likely two reasons for the predictive accuracy of these models. The first is that other than SC, GP1 and GP2 are the simplest models considered here, and in general simpler models are less likely to overfit the training data.[121] A similar trend of simpler models demonstrating greater extrapolative ability was observed by Zuo et al. in a recent comparison of different types of machine learned potential models.[50] The second reason is that these models were discovered in a hypothesis space designed to contain models resembling those for which there is fundamental physical justification. In general, the more physics can be included in the machine learning procedure, the more likely it is that a model will have extrapolative predictive power.

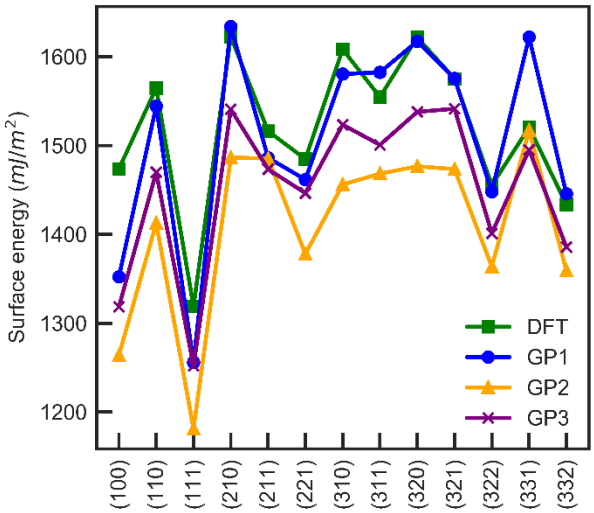


Figure 9. Surface energies of elemental copper as computed using DFT, and the interatomic potentials GP1, GP2, and GP3.

Genetic programming can use high-performing models that have been previously learned to seed a new search as a way of exploiting prior knowledge. This approach can be used

for finding better models with the same training dataset or with new training data. To demonstrate this approach, we have performed an additional search using an augmented training set in which the 13 low-index surfaces (shown in Figure 9) were added to the training data and always included in the subsets of data used to evaluate candidate models. This search was seeded with GP1 and GP2, and as a result the functional forms of the interatomic potential models that it discovered (Table 13) had many features in common with them. One of these models, which we label GP3 (equation (2.22)) resembles GP2 but, as expected, demonstrates better performance on surface energies (Figure 9). The absolute error for the weighted surface energies is 7% for GP3, as compared to 13% for GP2. The equation for GP3 is slightly simpler than that of GP2. The model GP3 is:

$$7.51\sum r^{3.98-3.93r} f(r) + \left(28.01 - 0.03\sum r^{11.73-2.93r} f(r)\right)\left(\sum f(r)\right)^{-1} \qquad (2.22)$$

Table 13. The 3-dimensional convex hull of models found by seeding with GP1 and GP2 and including the 13 low-index surfaces in the training data

Fitness	Cost*	Complexity	Expression
5404837	1	2	$\sum rf(r)$
1785.2	1	4	$-55.16\left(\sum rf(r)\right)^{-1}$
98.18	1	8	$\sum(669.45r^{-9.83}-0.10)f(r)$
58.56	1	10	$\sum(r^{10.21-5.48r}-0.07)f(r)$
55.35	4	13	$\left(1431.13\left(\sum f(r)\right)\left(\sum r^{-12.95}f(r)\right)\right)^{0.02\sum f(r)}-0.12\sum f(r)$
10.54	2	15	$\sum r^{10.21-5.47r}f(r)+0.98\left(\sum 0.31^{r}f(r)\right)^{-1}$
9.08	2	21	$2.74\sum r^{7.36-4.86r}f(r)+29.59\left(\sum r^{3.61-1.19r}f(r)\right)^{-1}$
9.03	2	25	$\sum\left(2.75r^{7.36-4.86r}-0.0009\right)f(r)$ $+29.73\left(\sum r^{3.61-1.19r}f(r)\right)^{-1}-15.98$
5.98	3	26	$7.51\sum r^{3.98-3.93r}f(r)+\left(28.01-0.03\sum r^{11.73-2.93r}f(r)\right)\left(\sum f(r)\right)^{-1}$
5.97	4	31	$\sum\left(7.50r^{3.98-3.93r}+0.001\right)f(r)$ $+\left(28.5-0.03\sum r^{11.73-2.94r}f(r)\right)\left(\sum f(r)\right)^{-1}$
5.54	4	39	$7.03\sum r^{4.00-3.88r}f(r)+\left(26.18-0.03\sum r^{11.73-2.94r}f(r)\right)\left(\sum f(r)\right)^{-1}$ $+0.92\left(\sum r^{3.50-1.52r}f(r)\right)^{-1}-136.85$

Notes: the model with fitness 5.98 is GP3. "Cost" is based on the number of summations. $f(r)$ is the smoothing function.

On average, GP3 and GP2 have a similar performance on the other properties listed in Table 6, even though GP3 has a better performance on surface energies. GP3 performs slightly worse on average on elastic constants and phonon frequencies, but significantly better on the dumbbell formation energy and stacking fault energies. It is difficult to assess the extent to which these changes in performance can be attributed to the addition of surfaces to the training data due to the stochastic nature of the search.

Although GP1, GP2 and GP3 are simpler than many other EAM-type models, they have a similar computational cost when implemented in LAMMPS[8, 137] due to the extensive use of tabulated values. Based on our benchmarks (Figure 12) GP1 takes 2.1 μs/step/atom, GP2

3.5 μs/step/atom, and GP3 takes 3.6 μs/step/atom, whereas EAM1 has a cost of 3.0 μs/step/atom. These speeds rank them among the fastest potential models, capable of modeling systems at large time and length scales.[8]

2.4.4 Analysis of the functional form of GP3

The simplicity of GP1, GP2 and GP3 makes them more interpretable than other machine learning interatomic potentials. For example, GP3 has the following form:

$$E_i = 7.51\sum_j r_{ij}^{3.98-3.93r_{ij}} f(r_{ij}) + \left(28.01 - 0.03\sum_j r_{ij}^{11.73-2.93r_{ij}} f(r_{ij})\right)\left(\sum_j f(r_{ij})\right)^{-1} \quad (2.23)$$

where E_i is the local energy around the i^{th} atom, r_{ij} is the distance between the i^{th} atom and its j^{th} neighbor, and $f(r_{ij})$ is a smoothing function that goes to zero at the cutoff radius.

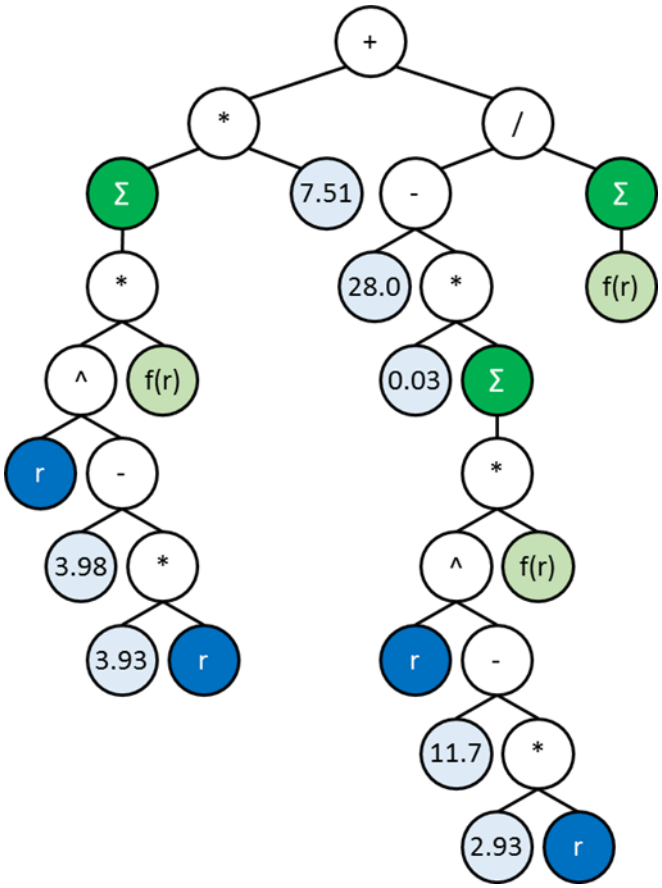


Figure 10. Tree representation of GP3.

The form of GP3 resembles that of the embedded atom model, with a pairwise repulsive term and a many-body attractive term formed by a non-linear transformation of pairwise interactions (Figure 11). The attractive term has an unusual form for an embedded atom model, depending on what is effectively a weighted average of an attractive potential, with the smoothing function providing the weights. The accuracy of this potential for a variety of physical properties suggests that this form might have underlying physical justification that may be useful in the construction of new potentials.

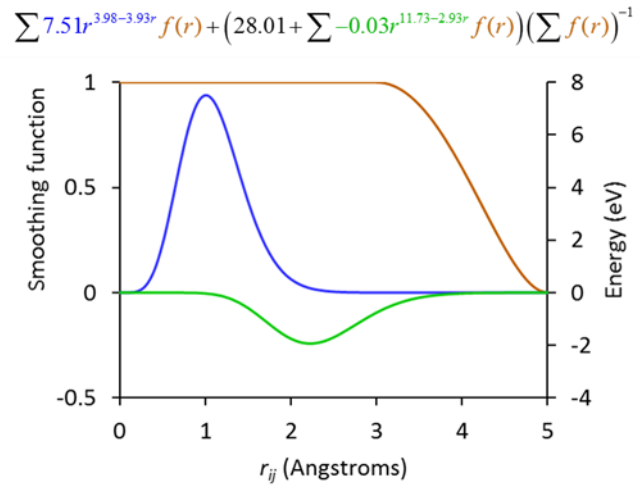


Figure 11. Different components of the potential model GP3. A repulsive interaction (right axis) is shown in blue, the attractive interaction (right axis) in green, and the smoothing function (left axis) in brown.

2.4.5 Benchmarks of computational cost

The interatomic potentials GP1, GP2 and GP3 were implemented in LAMMPS. The performance was measured in a system with 32 atoms for 10 million relaxation steps with a timestep of 1 fs in a single core. The paper by Sutton and Chen does not report a cutoff radius, but we found that a radius of at least 10 Å was required to reproduce their results. If we use the same 5 Å radius as used for GP1 and GP2, Sutton Chen EAM and GP1 have similar speeds because both are EAM-type potentials with 2 summations. EAM1 is slower than GP1 (Figure 12) because it used a greater cutoff distance of 5.50679 Å. GP2 is slower because it has 3 summations. However the difference in speed between the potentials compared in Figure 12 is small compared to the difference in speed between EAM and other potential models, which can be several orders of magnitude.[8, 153]. The computational cost was measured on a single core of a Haswell node with a clock speed of 2.5 GHz. The

benchmarking simulation consisted of 10,000,000 molecular dynamics steps for a 32-atom unit cell.

2.5 Discussion and conclusion

There are various machine learning approaches for developing interatomic potential models, and all of them have advantages and disadvantages. The general idea in many machine learning approaches is to construct a highly flexible hypothesis space that respects local symmetry and, with the help of large amounts of training data, identify the models within that hypothesis space that best reproduce the training data. Some examples of interatomic potential models that follow this general idea include (but are not limited to) neural network potentials, Gaussian approximation potentials, moment tensor potentials, SNAP potentials, and AGNI force fields. [54, 56-58, 71] Such models are capable of achieving very high accuracy for systems in which the local environments of the atoms are similar to those contained in the training set. These machine learning algorithms typically produce potential models that are orders of magnitude faster than DFT but also orders of magnitude slower than EAM-type potentials.[8, 51, 59, 150, 154]

In this chapter, we have demonstrated that machine learning can be used to develop simple and fast interatomic potential models from DFT training data, which facilitate simulations of systems at extreme time and length scales. The key to our approach is to search for parsimonious and computationally fast models in a hypothesis space that is constructed so that it contains simple models that are also physically meaningful and performing the learning process with genetic programming. Then we select the models based on a combination of simplicity, speed, and accuracy relative to the training data. The use of simplicity as a selection criterion results in models that are more likely to generalize well,

and it also significantly reduces the amount of data required to train the model.[155-157] For example, GP1 and GP2 were trained with 75 32-atom structures, for a total of 2400 atomic environments. For comparison, Artrith and Behler[151] have constructed a neural network potential for copper with a focus on surfaces. The potential was trained using 554,187 atomic environments, including tens of thousands of slabs and cluster structures. The neural network potential performs comparably to GP1 and GP2 for many bulk properties, and much better for surface energies (Table 8). The neural network approach demonstrates very low errors on the types of systems on which it was trained, but as the genetic programming approach requires less training data it is likely that some accuracy can be gained by using more accurate (and computationally expensive) methods to generate the training data.

EAM-type models are among the fastest interatomic potential models. The potential models discovered by the genetic programming approach are as fast as EAM-type models and demonstrate good predictive accuracy on properties they were not trained on. The performance of GP1 and GP2 in predicting surface energies is surprisingly good. The mean absolute error of surface energies predicted by GP1 is only 35mJ/m^2, even though there were no surfaces in their training data. The functional forms learned by the genetic programming algorithm, which was trained only on DFT data, resemble widely-used glue potentials with a unique form for the many-body term that depends on the inverse of a sum over pair interactions. Generating potential models using simple analytical expressions has several advantages, and one of them is that it may be possible to analyze the expressions to get an insight into the underlying physical interactions that are responsible for the shape of the potential energy surface.

The approach presented in this chapter has some notable areas of improvement and limitations. When developing interatomic potential models for different systems, it will be necessary to ensure that the hypothesis space contains simple expressions that capture important contributions to the potential energy. As an example, introducing terms that depend on bond angles are likely necessary for many new systems, and this was not done in this thesis. Another improvement to our approach would be to let the inner and outer cutoff radii vary in the way other parameters are optimized; in this study we used fixed inner and outer cutoff distances. There is also the question of how to determine which of the models discovered by the genetic programming algorithm provide the best balance of speed and predictive accuracy. This could be achieved in a number of ways,[88, 90, 158] including by evaluating performance against validation data (as done here) but it is not clear which approach is best. Finally, the genetic programming approach is likely not suitable for on-the-fly learning. Because it is a stochastic method, it can take an indeterminate amount of time to find a set of promising models, and there is no guarantee that an incremental change to the training data will result in an incremental change to the shapes of the potential energy surfaces on the convex hull. Other potential model approaches are probably better-suited for this purpose. Despite these current limitations, our results demonstrate that machine learning holds great promise to improve the accuracy of atomistic calculations at extreme time and length scales.

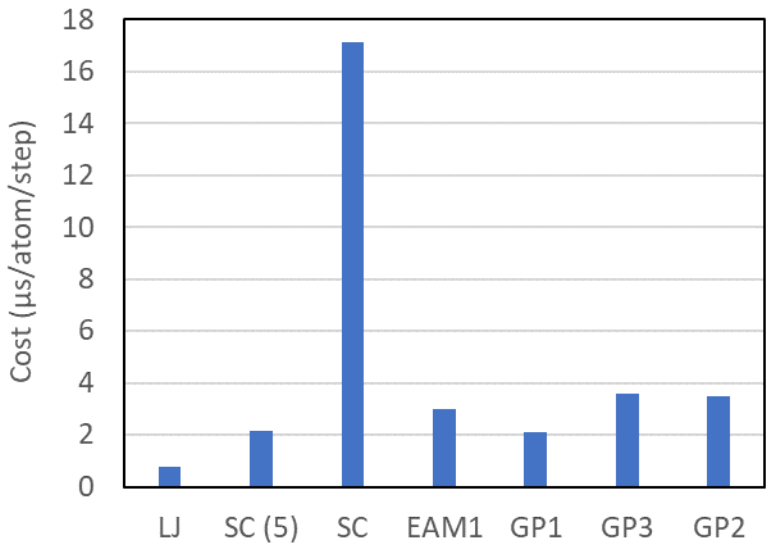


Figure 12. Computational cost of potential models in LAMMPS. SC (5) uses a cutoff distance of 5 Å, SC uses a cutoff distance of 10 Å. The cost is similar for SC (5), EAM1, GP1, GP2 and GP3.

2.6 Data availability

Our code is open source and available at https://gitlab.com/muellergroup/poet. The instructions for using GP1, GP2 and GP3 on LAMMPS are provided in the Methods section, and the files LAMMPS tables can be provided upon request.

3 Generalizability of the Functional Forms of Interatomic Potentials Discovered using POET

3.1 Introduction

Researchers across several fields apply molecular dynamics and Monte Carlo simulations to advance the scientific understanding, discovery, and design of materials and molecules. Using these methods, the thermodynamic state and kinetic behavior of a material can be computed with knowledge of the potential energy surface. *Ab initio* methods such density functional theory [149] (DFT), which are accurate across many chemistries and configurations of atoms [159], can be used to compute the potential energy surface, but the computational cost of these methods severely limits the time and length scales that can be practically modeled. Surrogate models, such as cluster expansions [107] and interatomic potential models (or force fields), [36-37, 69, 82, 114, 118, 160-163] are normally orders of magnitude faster than *ab initio* methods and scale better with respect to system size. The improved speed and scaling of surrogate models enable atomistic simulations that inform the design of materials at larger time and length scales.

Classical (or empirical) interatomic potential models [24, 29-31, 110-111, 118], like the embedded atom method [110] (EAM), are usually derived from physical principles, which can make them transferable to atomic configurations that are significantly different from those in their training set. However they have limited accuracy due to their fixed functional forms. In the last decade, researchers have made great progress in the development of accurate interatomic potential models via supervised machine learning, [50, 53-54, 56-58, 60, 64, 69, 71, 74, 77, 79, 81-82, 89, 102, 104, 106-108, 164-165] but their computational speed is typically 2-3 orders of magnitude slower than classical interatomic potentials[50]. In addition, the complexity of

many machine-learned interatomic potentials may lead to poor transferability. There has recently been progress in addressing this issue, but it remains an important challenge in the field. [49, 166-168]

Our research group previously demonstrated the use of supervised machine learning to develop accurate many-body interatomic potential models for copper that have simplicity, speed and transferability that were comparable to, and in some cases better than, classical interatomic potential models. [89] We accomplished this by using symbolic regression in the form of genetic programming to explore a hypothesis space of simple models designed to have physically meaningful terms. The good transferability that these models presented suggests that their functional forms encode underlying physics. Our algorithm was implemented in the open-source software package Potential Optimization by Evolutionary Techniques (POET) (https://gitlab.com/muellergroup/poet.git).

Here, we extend our analysis of this approach to the fcc transition metals from groups 9, 10 and 11 on the periodic table: Cu, Ag, Au, Ni, Pd, Pt, Rh, and Ir. We demonstrate that the functional forms discovered for copper in our previous work (named GP1, GP2, and GP3) [89] generalize well to elemental systems that are chemically similar to Cu, like Ag, Ni, Pd, and Pd, and that POET can find new, accurate functional forms for these elemental systems. Using the Sutton Chen [138] functional form as a benchmark due to its simplicity and physical foundations, we show that POET models tend to have much lower errors with similar complexity. When compared against EAM-type interatomic potential models from the literature, the models developed using genetic programming in this chapter have less error on the validation properties than about 50% of the literature models, and they are on average more than 10 times simpler.

3.2 Methods

3.2.1 Developing the interatomic potential models

In our previous work, [89] we used POET to develop new functional forms for interatomic potential models for Cu; shown in equations (3.1), (3.2), and (3.3). In the present chapter, we optimized the parameters of these functional forms for Cu, Ag, Au, Ni, Pd, Pt, Rh and Ir. The objective function was the fitness shown in equation (3.4). We re-optimized the parameters for Cu for methodological consistency. We used the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [126] restarting with increasing population size (IPOP-CMA-ES), [169] and the conjugate gradient optimizer [127]. We used the population sizes of 10, 30, 50, 70, 90 and 110, and after each CMA-ES run, we optimized the parameters with the conjugate gradient optimizer. We executed a run of IPOP-CMA-ES and conjugate gradient for each model and each element, for a total of 24 runs. We then took the model with the best training fitness for each element and functional form. The models obtained in this way are labeled GP1-c, GP2-c, and GP3-c (Table 14).

$$E_i = \sum_j (r^{x_0 - x_1 r} - x_2^{\,r}) f(r) + x_3 / \sum_j x_4^{\,r} f(r) \tag{3.1}$$

$$E_i = x_0 \sum_j r^{x_1 - x_2 r} f(r) + \left(x_3 - \sum_j \left(x_4 + x_5 r^{x_6 - x_7 r} \right) f(r) \right) / \sum_j f(r) \tag{3.2}$$

$$E_i = x_0 \sum_j r^{x_1 - x_2 r} f(r) + \left(x_3 - x_4 \sum_j r^{x_5 - x_6 r} f(r) \right) / \sum_j f(r) \tag{3.3}$$

On equations (3.1), (3.2), and (3.3), E_i is the energy associated with an atom, Σ is the summation over neighbors of atom i, and r is the distance between atom *i* and atom *j*, the parameters are denoted by x, and $f(r)$ is a smoothing function [122]. The fitness is:

$$fitness = 0.5MSE_{energy} + 0.4MSE_{force} + 0.1MSE_{stress} \quad (3.4)$$

where MSE_{energy} is the mean squared error of the normalized energies, and the normalization was done by subtracting the minimum energy and dividing by the standard deviation of the energies, MSE_{force} is the mean squared error of the normalized components of the force vectors, and the normalization was performed by subtracting the mean of the forces and dividing by the standard deviation of the forces, finally MSE_{stress} is the mean squared error of the normalized components of the stress tensor, where normalization was done in the same way as for the forces described here.

Table 14. Acronyms of the interatomic potential models discussed in this chapter.

Acronym	Description
GP1	Model for Cu[89]
GP2	Model for Cu[89]
GP3	Model for Cu[89]
GP1-c	Optimized the parameters in GP1 [89] with IPOP-CMAES and conjugate gradient
GP2-c	Optimized the parameters in GP2 [89] with IPOP-CMAES and conjugate gradient
GP3-c	Optimized the parameters in GP3 [89] with IPOP-CMAES and conjugate gradient
SC4-c	Optimized the parameters in Sutton Chen with IPOP-CMAES and conjugate gradient, but maintained the square root
SC5-c	Optimized the parameters in Sutton Chen with IPOP-CMAES and conjugate gradient, allowing the power of 0.5 to change
GPn	New functional forms introduced in this chapter, discovered using POET, for Cu, Ag, Au, Ni, Pd, Pt, Rh and Ir.

To compare the performance of our models against a simple and physics-derived interatomic potential model, we optimized the parameters of the Sutton Chen [138] model with IPOP-CMA-ES and conjugate gradient starting from known parameters for each fcc element (Table 15). Equation (3.5) shows the Sutton Chen EAM model, where x_i are the parameters. In the Sutton Chen EAM model, x_4 is 0.5. For each element, we ran an optimization of a Sutton Chen model maintaining the square root, and we call these models SC4-c, and we ran an optimization allowing x_4 to change and call these models SC5-c, for a total of 16 runs (Table 14).

$$E_i = \sum_j x_0 r^{x1} f(r) - \left(\sum_j x_2 r^{x_3} f(r) \right)^{x_4} \tag{3.5}$$

Table 15. Initial parameters of SC models [138]

element	x0	x1	x2	x3	x4
Cu	644.524255	-9	22.97001139	-6	0.5
Ag	27844.52505	-12	25.11061249	-6	0.5
Au	8176.059345	-10	121.9754648	-8	0.5
Ni	651.549205	-9	27.01282713	-6	0.5
Pd	25086.3566	-12	52.52960227	-7	0.5
Pt	8496.08965	-10	161.1358259	-8	0.5
Rh	22379.22732	-12	39.12190388	-6	0.5
Ir	185600.0961	-14	46.44422713	-6	0.5

Note: The parameters of SC correspond to equation (3.5)

Finally, we used POET to discover new functional forms for each of the transition fcc elements on groups 9, 10, and 11 (Table 14 and Table 22). We run the search using 5 different initializations: seeding with GP1, GP2, or GP3 from our previous work [89], seeding with a Sutton Chen model, or starting from randomly generated functions using the grow or full methods [100] with equal probability. We ran 30 optimizations with each of these

starting configurations, for each element, and for two kinds of smoothing functions, for a total of 2400 runs. One smoothing function is from [122], which we used in our previous work [89] and the other is shown on equation (3.7). We added this smoothing function because its second derivative is zero at the inner and outer cutoff distances.

Each 30 runs for every element and seed of a POET run resulted in a set of interatomic potential models. For each element, we selected a single model by combining the sets of models for that element and building a 3-dimensional convex hull of complexity, computational speed, and training fitness. Then, using the 3-dimensional convex hull for each element, we created a 2-dimensional convex hull using a selection value and complexity, and then chose the model at the elbow on this hull (Figure 13 to Figure 20). We defined the elbow using a slope of less than -0.005. We defined the selection value as the average of the normalized fitness and the normalized MAPE on elastic constants (equation (3.6)). The normalization of the fitness was done by taking the natural logarithm of the training fitness value, and then doing a min-max normalization. For the MAPE on elastic constants, we directly performed a min-max normalization. The complexity was measured as the number of nodes in the tree-graph representation of the potential model, the computational speed was measured as the number of summations over neighbors, and the fitness as the weighted average of the normalized mean squared errors on energies, forces and stresses with respect to DFT training data (equation (3.4)).

$$s_i = \frac{1}{2}\left(\frac{\ln fitness_i - \min \ln fitness}{\max \ln fitness - \min \ln fitness} + \frac{MAPE_{EC,i} - \min MAPE_{EC}}{\max MAPE_{EC} - \min MAPE_{EC}} \right) \quad (3.6)$$

where s_i is the fitness of model “i" and $fitness_i$ is the fitness of model “i" defined in equation (3.4).

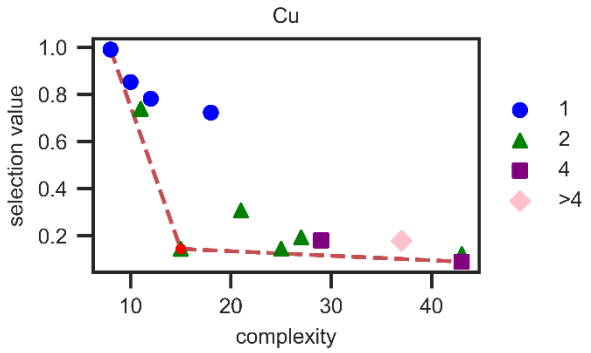


Figure 13. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Cu. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model.

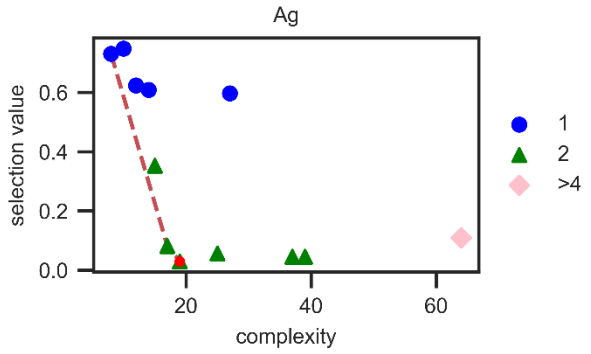


Figure 14. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Ag. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model.

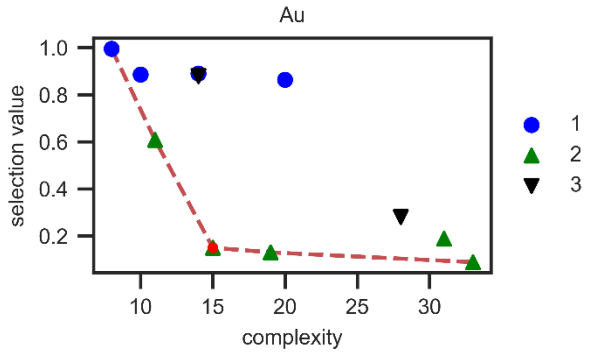


Figure 15. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Au. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model.

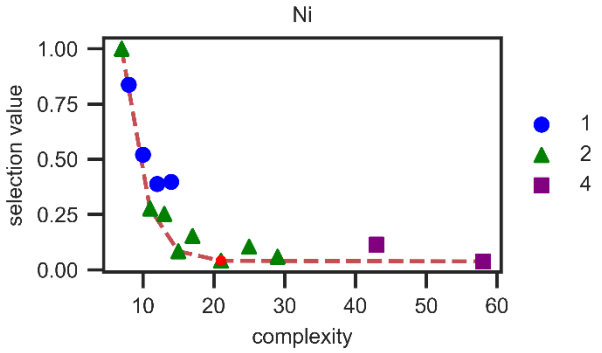


Figure 16. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Ni. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model.

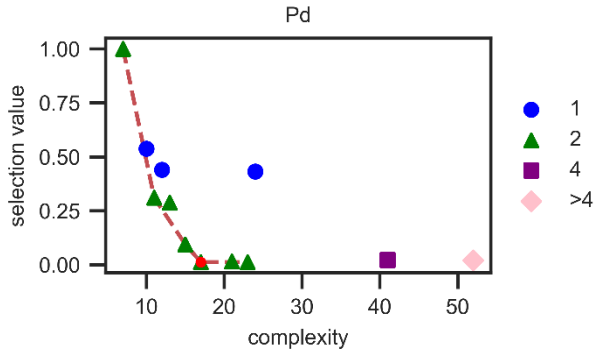


Figure 17. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Pd. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model.

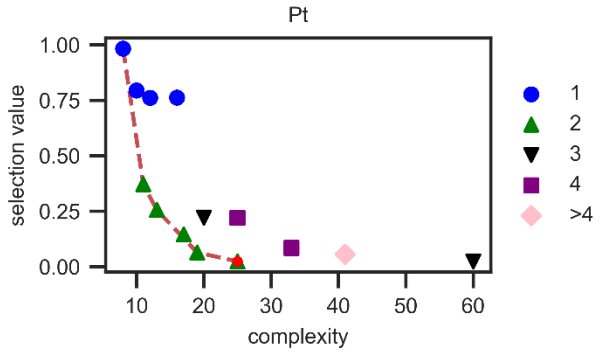


Figure 18. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Pt. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model.

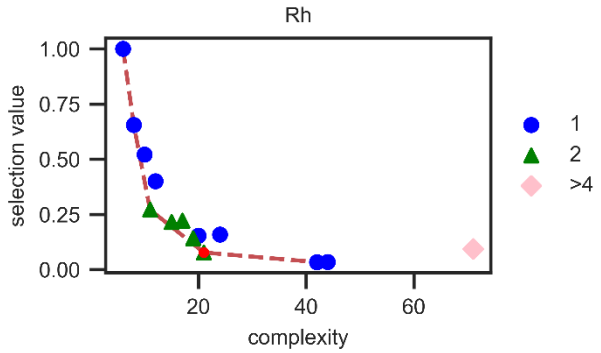


Figure 19. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Rh. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model.

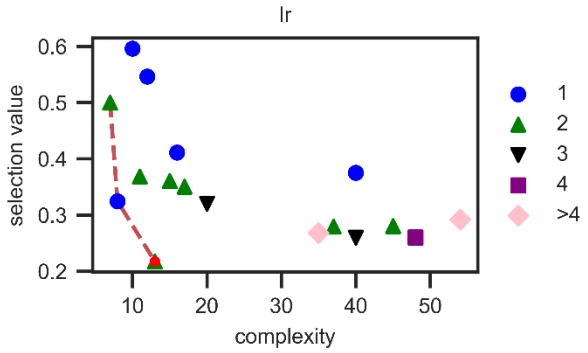


Figure 20. Convex hull of "selection value", see equation (3.6), and complexity (i.e., number of nodes) for selecting the model GPn for Ir. The convex hull is shown as the red dashed line, and GPn is shown as a red dot. The legend indicates the speed (i.e., number of summations over neighbors) of each interatomic potential model.

3.2.2 Density functional theory data generation

The DFT data were computed using the Vienna Ab initio Simulation Package (VASP) with the Perdew-Burke-Ernzerhof [129] (PBE) generalized gradient approximation (GGA) exchange correlation functional. The following projector augmented wave method [130] (PAW) pseudopotentials were used: Cu_pv, Ag, Au, Ni_pv, Pd, Pt, Rh_pv, and Ir (Table

16). Efficient k-point grids were obtained from the k-point grid server with MINDISTANCE > 50Å. [131] A plane-wave cutoff energy of 750 eV and ADDGRID = TRUE in VASP were used. The DFT point defect energies were computed by linear extrapolation. [132] The phonon dispersion curves were computed on a 3×3×3 supercell. The radial distribution function molecular dynamics simulations were performed in the NVT ensemble at the experimental the liquid density on a 3×3×3 supercell. To calculate the radial distribution function, DFT molecular dynamics was performed with a cutoff energy of at least 400 eV, the electronic self-consistency convergence was 10-5 eV, and only the k-point at Γ was used.

Table 16. Pseudopotentials used in VASP.

Species	Name (TITEL from POTCAR)
Cu	PAW_PBE Cu_pv 06Sep2000
Ag	PAW_PBE Ag 02Apr2005
Au	PAW_PBE Au 04Oct2007
Ni	PAW_PBE Ni_pv 06Sep2000
Pd	PAW_PBE Pd 04Jan2005
Pt	PAW_PBE Pt 04Feb2005
Rh	PAW_PBE Rh_pv 25Jan2005
Ir	PAW_PBE Ir 06Sep2000

The DFT training data for copper was taken directly from[89]. For the other elements we used a similar approach. We ran 3 types of DFT molecular dynamics simulations: NVT at 300K on the fcc phase, NVT at a temperature between the melting temperature and the boiling

temperature using the liquid density, and NPT at 300K starting from the fcc phase (Table 17). For each element, the training set had 75 structures, 75 energies, 7200 components of force, and 450 components of the virial stress. In total, we collected 50 snapshots from each molecular dynamics simulation. We took the first 25 snapshots from each simulation as the training set, and the last 25 snapshots from each simulation as a part of the validation set. For the computation of the fitness when fitting the models, the energies were transformed by subtracting the minimum and dividing by the standard deviation, and the forces and stresses were standardized by subtracting the mean and dividing by the standard deviation.

Table 17. Temperatures of the DFT molecular dynamics simulations used for generating the training and validation data.

Element	Temperature of NVT fcc (K)	Temperature of NVT liquid (K)	Temperature of NPT (K)
Cu	300	1400	1400
Ag	300	1535	300
Au	300	1637	300
Ni	300	2457	300
Pd	300	2742	300
Pt	300	3070	300
Rh	300	3354	300
Ir	300	4078.5	300

3.2.3 Computing properties with interatomic potential model

The data used to validate the interatomic potential models developed in this chapter were computed with LAMMPS. The supercell sizes were the same as the ones used for DFT calculations. Instructions and files required to use models on LAMMPS are provided on the Supplementary Information. The cutoff distances of Cu, Ag, Au, Ni, Pd, Pt, Rh and Ir interatomic potential models are on Table 18.

Table 18. Cutoff distances used for the interatomic potential models for each element.

Element	Outer cutoff distance, r_{out} (Å)	Inner cutoff distance, r_{in} (Å)
Cu	5	3
Ag	5.5	3.5
Au	5.5	3.5
Ni	5	3
Pd	5.5	3.5
Pt	5.5	3.5
Rh	5.25	3.25
Ir	5.4	3.4

3.3 Results and discussion

3.3.1 Assessing the transferability of functional forms developed with POET for Cu to other elemental systems.

The validation errors on energies, forces, and stresses of the interatomic potential models developed using the functional forms of GP1, GP2, and GP3 outperform Sutton Chen by an order of magnitude across the elements considered except Au where the validation fitness of SC5-c is of the same order of magnitude (Figure 21). This order of magnitude improvement comes with only a slight increase in complexity the number of nodes of the GP1-c, GP2-c, GP3-c functions and the Sutton Chen function are 19, 26, 24, and 15, respectively (Figure 41), and the models from genetic programming have 2 or 3 summations over neighbors. The SC5-c model for Au is particularly good compared to SC4-c and SC5-c models for other elements; it has a similar fitness as the GP1-c, GP2-c, and GP3-c models, which have reasonable validation MAEs: 6 meV/atom, around 90 meV/Å, and around 0.3 GPa.

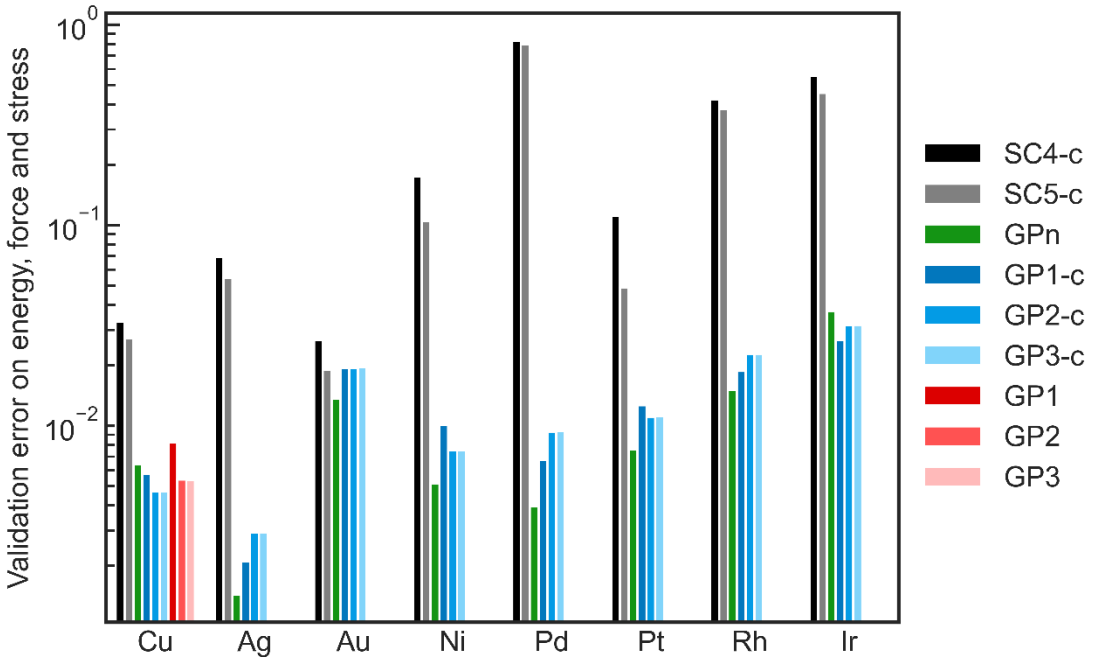


Figure 21. Error on the validation set of energies, forces and stresses in logarithmic scale with base 10. This error metric is the same as the fitness. The fitness is the weighted average of the normalized mean squared error on energy, force and stress, where the weights are 0.5, 0.4 and 0.1, respectively. The models are ordered in approximately increasing complexity. The GPn correspond to new functional forms developed with POET by seeding with the interatomic potentials of Sutton Chen, GP1, GP2, or GP3. The models SC4-c, SC5-c, GP1-c, GP2-c, and GP3-c were developed by optimizing the parameters of the corresponding functional forms using the CMA-ES and the conjugate gradient optimizer.

GP1-c, GP2-c and GP3-c for Cu are have a lower validation error on the energies, forces and stresses than the original models GP1, GP2, and GP3 discovered by POET in our previous work [89] (Figure 21). The parameters for these models were discovered using multiple runs of IPOP-CMA-ES and conjugate gradient, which is much more extensive than what is done in a typical search by POET, suggesting POET found local optima for the parameters. With more time or tighter convergence criteria, POET may be able to find functions that are more fit.

To assess the performance of the models derived from POET on a wider set of validation metrics, we calculated the average of the normalized error across the metrics: MAE of energies, MAE of forces, MAE of stresses, MAPE of C11, C12, C44, MAPE of 7 phonon frequencies, absolute percent error of vacancy formation energy, absolute percent error of vacancy migration energy, absolute percent error of dumbbell formation energy, MAPE of 13 low-index surface energies (except for GP3), absolute percent error of intrinsic stacking fault energy, absolute percent error of unstable stacking fault energy, absolute percent error of hcp formation energy, absolute percent error of bcc formation energy, absolute percent error of fcc lattice parameter, and absolute percent error of bcc lattice parameter. The normalization was done using min-max scaling (x-min(x))/(max(x)-min(x)). The models GP1-c, GP2-c, and GP3-c, which come from the functional forms GP1, GP2, and GP3, identified for Cu in our previous work [89], have good transferability on elements that are closer to Cu on the periodic table, such as Ag, Ni, and Pd. This finding suggests that the models GP1-c, GP2-c, and GP3-c encoded information about the physics of the Cu system that can be applied to other systems that are chemically similar to Cu. For all elements other than Au and Cu, the models discovered using genetic programming were significantly more accurate for the validation properties than the Sutton-Chen derived models (Figure 22), consistent with their validation performance on energies, forces, and virial stresses (Figure 21).

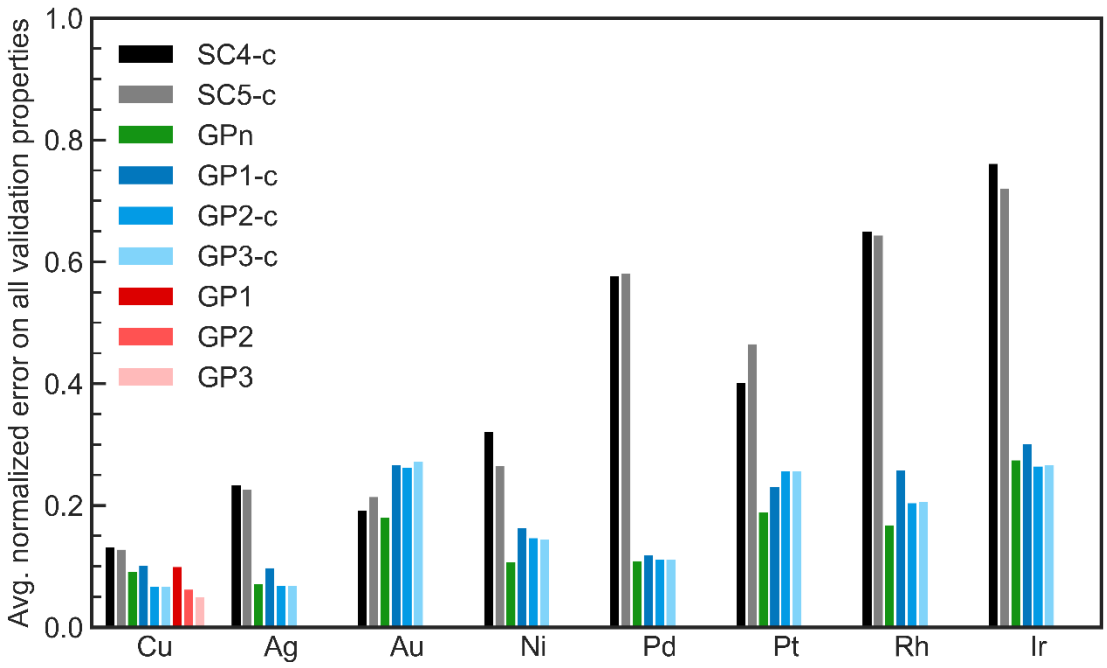


Figure 22. Average of normalized errors across validation properties. The validation metrics considered on this plot are: MAE of energies, MAE of forces, MAE of stresses, MAPE of C11, C12, C44, MAPE of 7 phonon frequencies, absolute percent error of vacancy formation energy, absolute percent error of vacancy migration energy, absolute percent error of dumbbell formation energy, MAPE of 13 low-index surface energies (except for GP3), absolute percent error of intrinsic stacking fault energy, absolute percent error of unstable stacking fault energy, absolute percent error of hcp formation energy, absolute percent error of bcc formation energy, absolute percent error of fcc lattice parameter, and absolute percent error of bcc lattice parameter. The normalization was done using min-max scaling (x-min(x))/(max(x)-min(x))

In general, the models with parameters discovered by POET for copper in our previous work [89] have lower errors than the models that have more extensively optimized parameters on the properties use to test transferability, suggesting that the more extensive search for optimal parameter values may have overfit the models. This difference is most notable between GP1-c and GP1, for which the parameters x_1 and x_3 in equation (3.1) decreased by 50% and 80%, respectively (Table 19). For GP3 and GP3-c (Table 21), some of this difference may be attributed to the fact that the training data for GP3 included 13 low-index

surfaces on its training data, but these surfaces were not used to train any of the other models, including GP3-c. Consequently, the validation properties of GP3 excluded the low index surfaces. It is possible that the inclusion of low index surfaces for developing GP3 gave it access to a wider range of atomic environments useful for predicting validation properties such as stacking fault energies, for which GP3 has an error 8 mJ/m^2 lower than GP3-c for Cu. GP2 has practically the same average normalized validation errors on all the validation properties as GP2-c, for which the MAEs are very close to each other (differences of 1 meV/atom, 3 meV/Å, and 0.04 GPa) but interestingly, the absolute percent change in the parameters is 53% (Table 20).

Table 19. Parameters of GP1 models

Element	x0	x1	x2	x3	x4
Cu [(a)]	10.213032	10.213032	0.210769	0.972441	0.328949
Cu	9.484145768	5.085386294	0.290921084	0.1915873	0.200459294
Ag	9.69475452	4.516520485	0.000333	0.183431686	0.235142007
Au	11.87322967	5.399305334	0.000638	6.614996057	0.742988891
Ni	10.41736297	5.551786863	0.302376677	0.148148977	0.157143032
Pd	10.10090728	4.761956156	0.306976252	0.172988701	0.203414297
Pt	11.70379336	5.391638705	0.000179	5.169953388	0.536014872
Rh	10.29114394	4.786653571	0.395012273	0.04272145	0.109993287
Ir	10.9739431	4.880921708	0.406723036	0.02778075	0.087663454

Notes: (a) Parameters from [89]. The parameters in GP1 correspond to: $E_i = \sum_j (r^{x_0 - x_1 r} - x_2{}^r) f(r) + x_3 / \sum_j x_4{}^r f(r)$

Table 20. Parameters of GP2 models

El.	x0	x1	x2	x3	x4	x5	x6	x7
Cu [(a)]	7.325665	3.979468	3.935002	27.319252	11.126676	0.034045	11.73571	2.926828
Cu	9.70892165	2.04216189	3.11153727	29.7973808	12.5423967	0.13057011	10.3213873	2.89972650
Ag	28.2108875	1.30043133	2.63967947	22.0842139	18.5193986	0.29691126	6.80073178	1.82397066
Au	35.5079481	3.37164794	3.52961596	20.1493869	7.26029194	7.83228003	-2.37951974	-0.0422891
Ni	5.560674959	5.771775656	4.807729452	35.62554898	2.008914012	0.000246	22.9884148	5.2725877
Pd	9.618326536	3.997224815	3.352722931	41.55663361	0.847535341	2.213072697	4.18967871	1.4505722
Pt	15.41217564	4.697571488	3.77267856	51.27691626	4.301949264	0.003134623	9.684177103	1.53515267
Rh	8.324596442	4.582036048	3.670327541	47.50099338	13.49894162	2.61E-05	25.05670522	5.13638985
Ir	1.665687389	9.82561501	4.890543442	63.83216173	6.352562049	1.64E-05	25.3949301	5.14740146

Notes: (a) Parameters from [89]. The parameters in GP2 correspond to:

$$E_i = x_0 \sum_j r^{x_1 - x_2 r} f(r) + \left(x_3 - \sum_j \left(x_4 + x_5 r^{x_6 - x_7 r} \right) f(r) \right) / \sum_j f(r)$$

Table 21. Parameters of GP3 models

El.	x0	x1	x2	x3	x4	x5	x6
Cu [(a)]	7.508311	3.979897	3.934521	28.013689	0.031791	11.734548	2.933153
Cu	9.653998788	2.060534748	3.116552313	29.79705838	0.131622327	10.30393221	2.896217937
Ag	26.4855544	1.467960106	2.679806316	22.02488054	0.316159588	6.676462684	1.801372235
Au	18.85588321	4.951473594	3.89542606	20.0709281	3.146357686	-0.925808305	0.003602991
Ni	7.48071736	4.684675392	4.471605354	35.92501217	0.000246	23.12325616	5.328349011
Pd	11.05054968	3.581177506	3.263151364	40.97222955	0.974045504	5.629805775	1.686345098
Pt	9.926992479	5.890721459	4.067027873	50.47696829	0.005335237	9.087390711	1.459929578
Rh	16.91561527	2.441522737	3.097197331	48.21641282	4.12E-05	24.67533329	5.140494131
Ir	1.312187465	10.27621494	4.942567634	65.27435096	0.000437	19.57546211	4.188192247

Notes: (a) Parameters from [89]. The parameters in GP3 correspond to:

$$E_i = x_0 \sum_j r^{x_1 - x_2 r} f(r) + \left(x_3 - x_4 \sum_j r^{x_5 - x_6 r} f(r) \right) / \sum_j f(r)$$

The parity plots of the models GP1-c, GP2-c and GP3-c (figures below) show that they are not overfitting the data, and they have lower mean absolute errors on validation energies, components of the force, and components of the stress than the SC and SC4-c models (Appendix A). The parity plots of models discovered in this work using POET, which we

call GPn and discuss later in the text, also show that they are not overfitting the data, and they have low mean absolute errors.

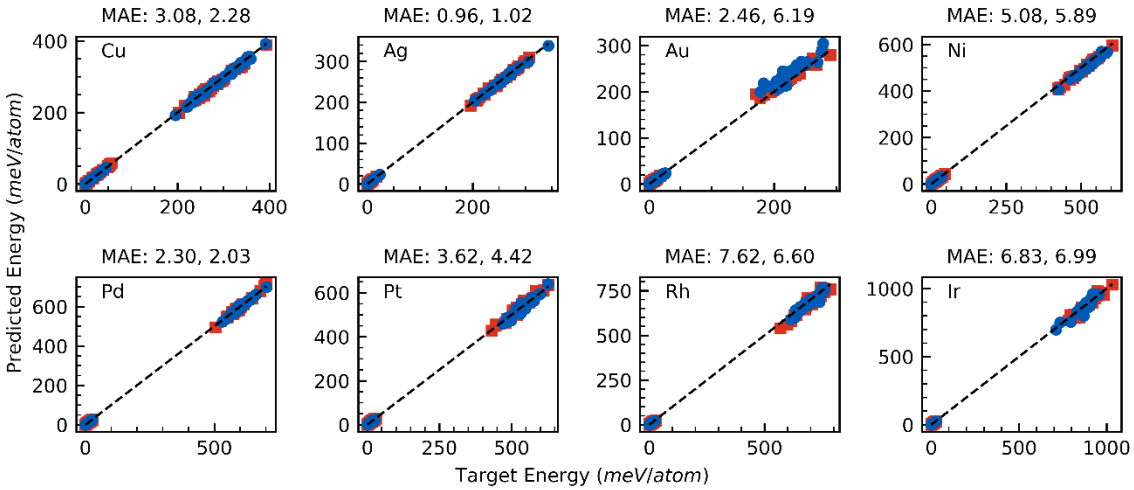


Figure 23. Mean absolute errors (MAE) of GP1-c models on energies in meV/atom. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

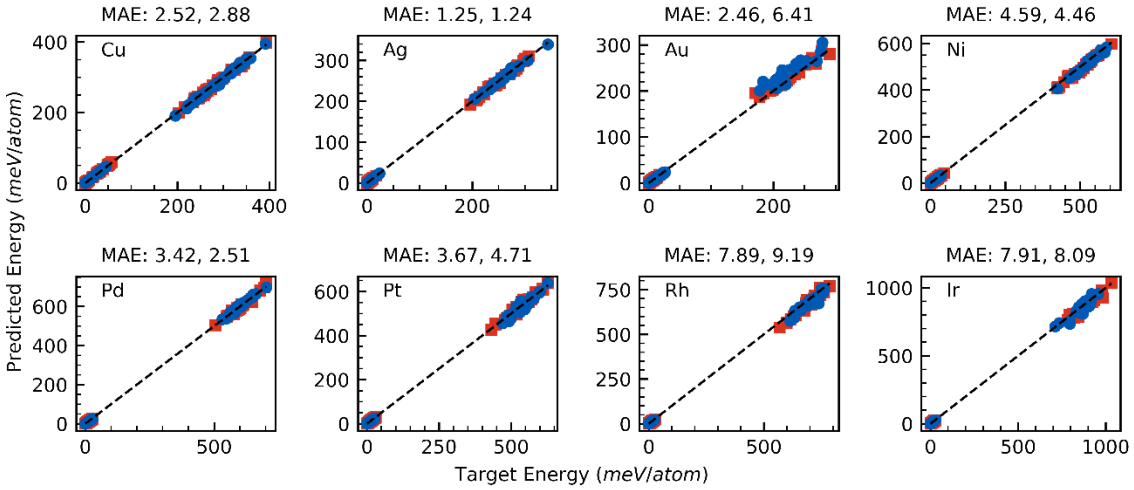


Figure 24. Mean absolute errors (MAE) of GP2-c models on energies in meV/atom. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

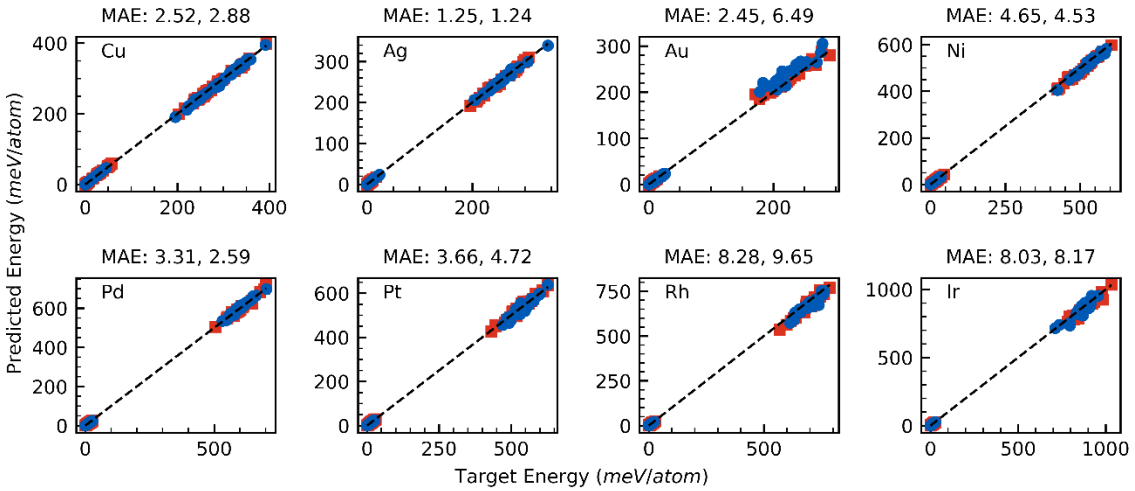


Figure 25. Mean absolute errors (MAE) of GP3-c models on energies in meV/atom. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

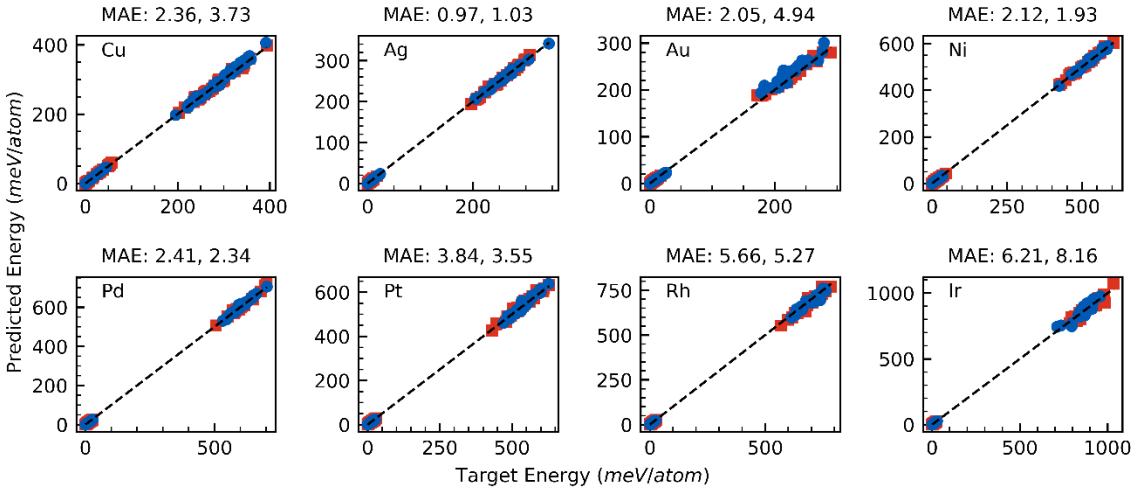


Figure 26. Mean absolute errors (MAE) of GPn models on energies in meV/atom. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

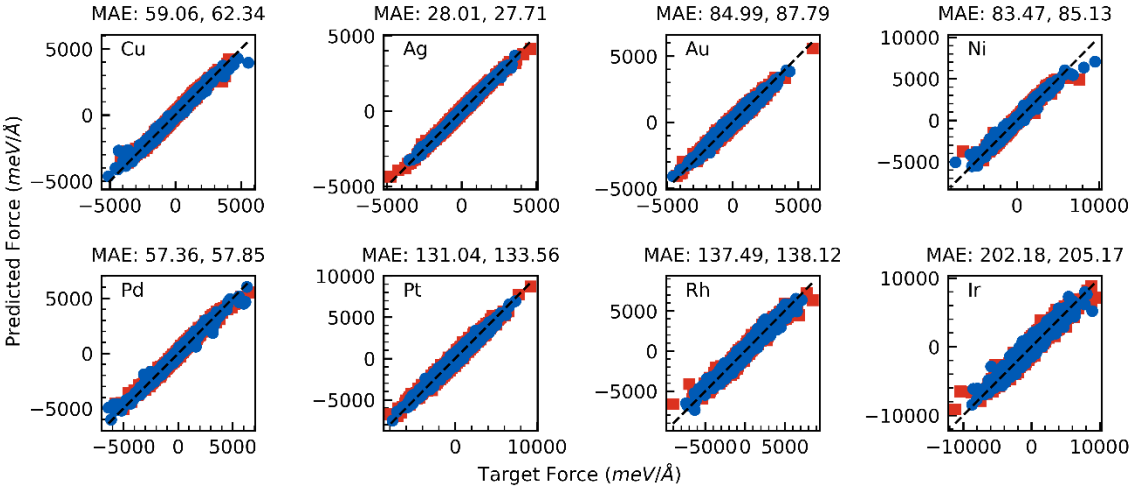


Figure 27. Mean absolute errors (MAE) of GP1-c models on the components of the forces in meV/Å. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

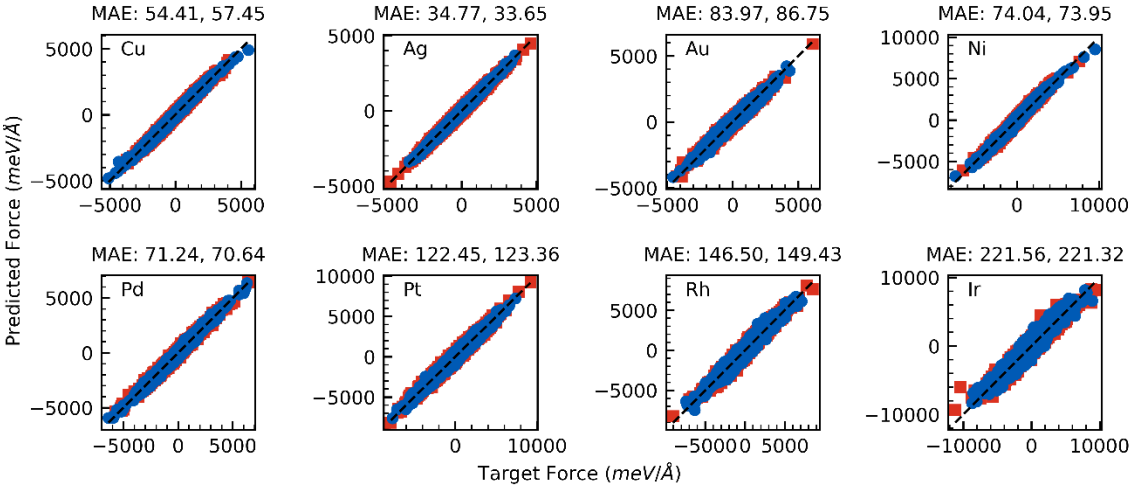


Figure 28. Mean absolute errors (MAE) of GP2-c models on the components of the forces in meV/Å. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

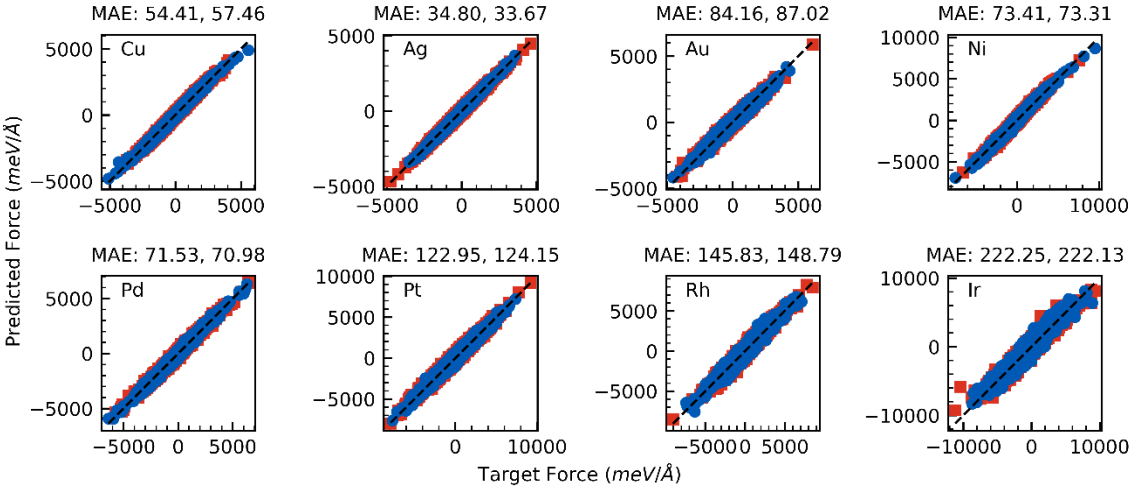


Figure 29. Mean absolute errors (MAE) of GP3-c models on the components of the forces in meV/Å. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

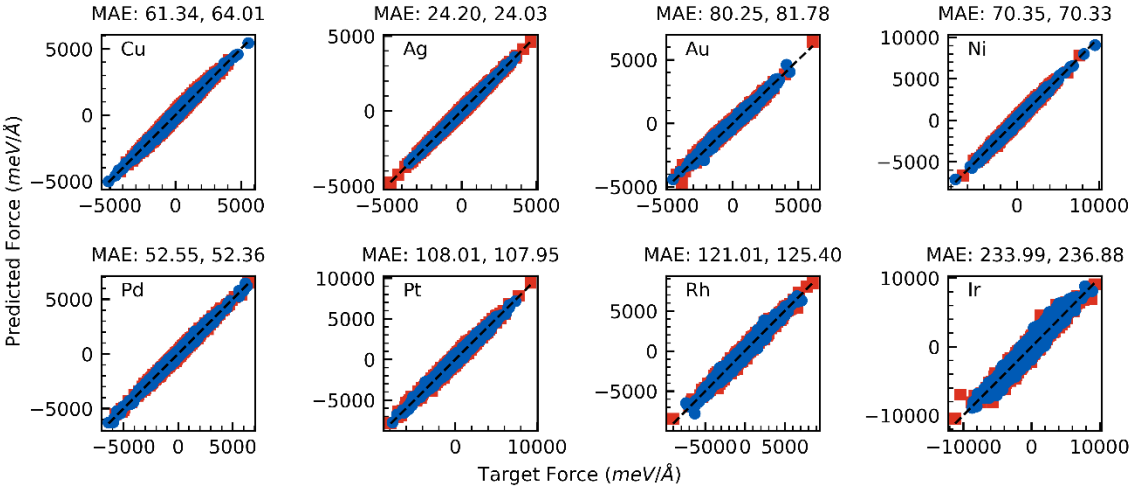


Figure 30. Mean absolute errors (MAE) of GPn models on the components of the forces in meV/Å. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

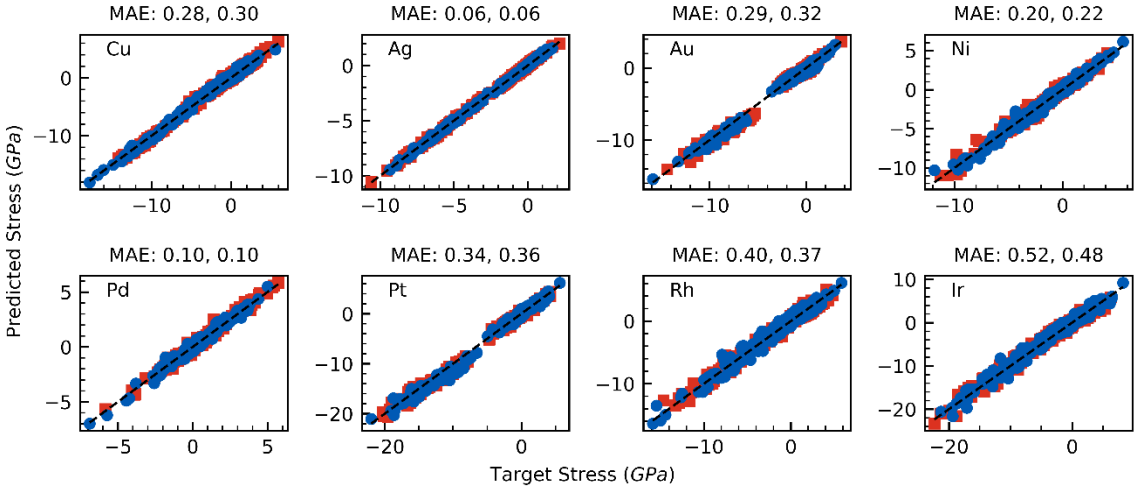


Figure 31. Mean absolute errors (MAE) of GP1-c models on the components of the virial stress tensor in GPa. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

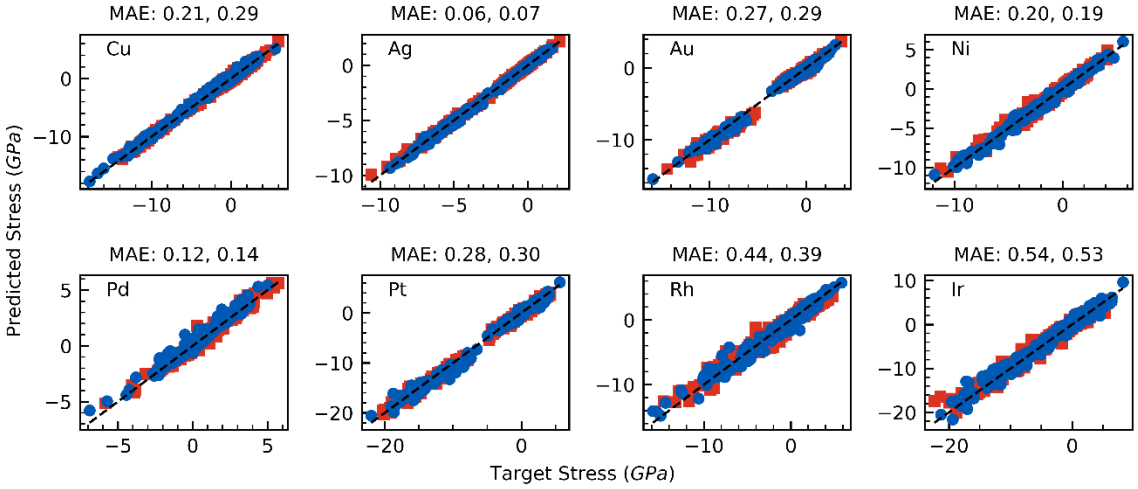


Figure 32. Mean absolute errors (MAE) of GP2-c models on the components of the virial stress tensor in GPa. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

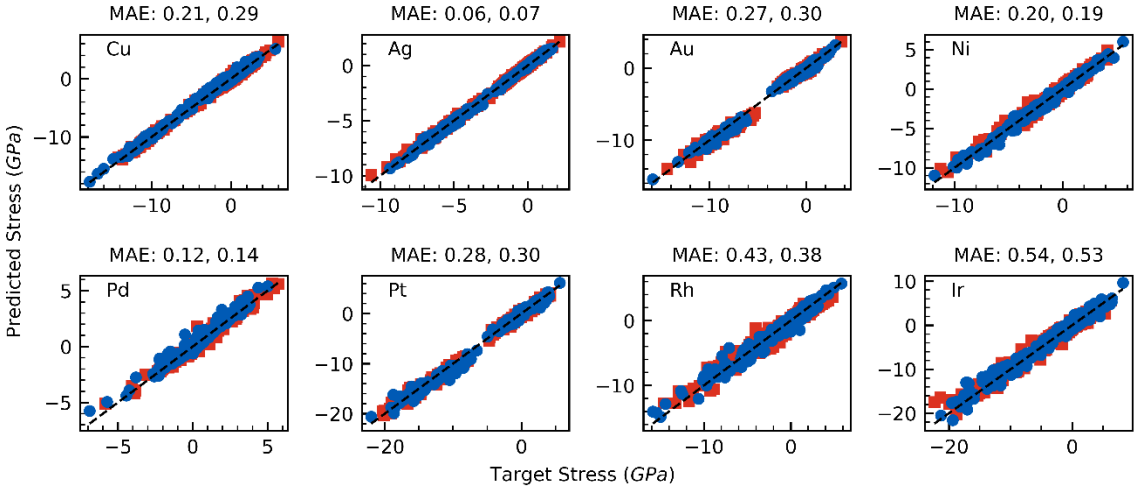


Figure 33. Mean absolute errors (MAE) of GP3-c models on the components of the virial stress tensor in GPa. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

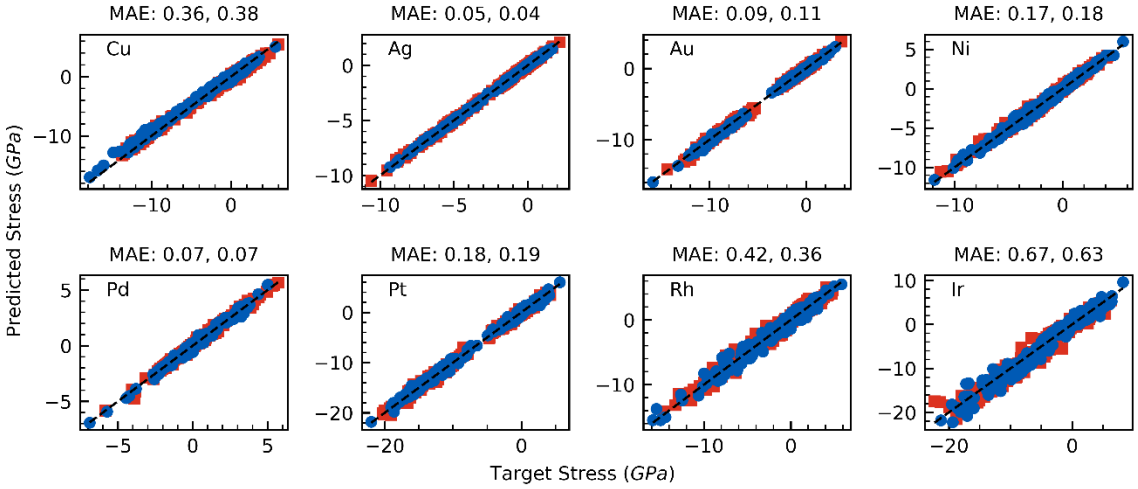


Figure 34. Mean absolute errors (MAE) of GPn models on the components of the virial stress tensor in GPa. The training points are red squares, and the validation points are blue circles. The first value on top of the plot is the training MAE, and the second is the validation MAE.

The box plots of the absolute errors on the energies (Figure 35), on the components of the force (Figure 36) and on the components of the virial stress tensor (Figure 37) for each element show that the models derived from genetic programming have a smaller spread in

the absolute errors (smaller inter quartile ranges) and a lower median, indicating that they achieve a better fit than SC4-c and SC5-c models across all the elements, except for the energies of SC4-c and SC5-c for Au, which have a slightly better performance as the models developed using genetic programming.

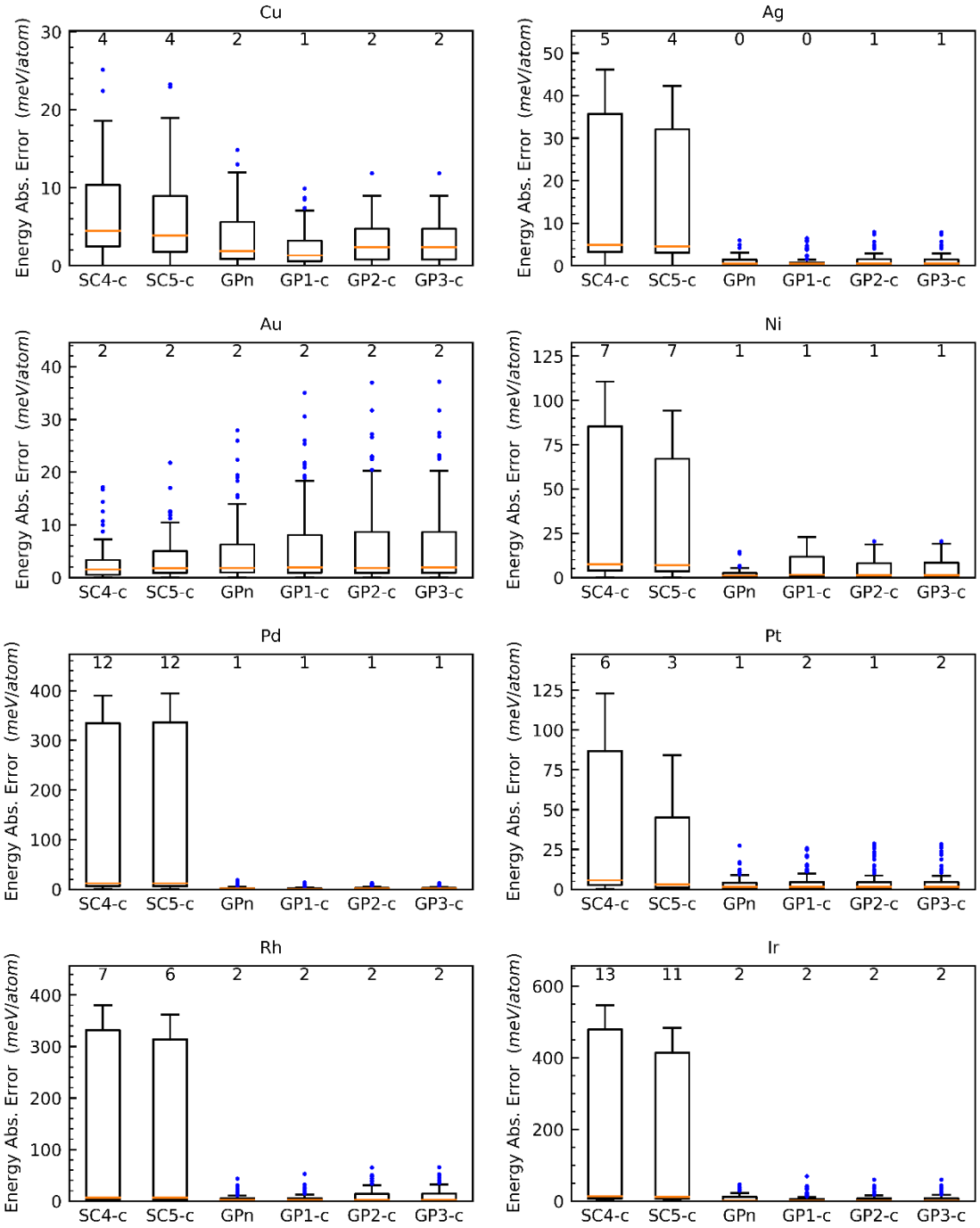


Figure 35. Box plots of absolute errors on the validation energies. The orange lines are the median, and the values of the medians are shown on top. The boxes show the interquartile range (IQR), and the whiskers are at 1.5 of the IQR, the blue points are the outliers beyond 1.5 of the IQR.

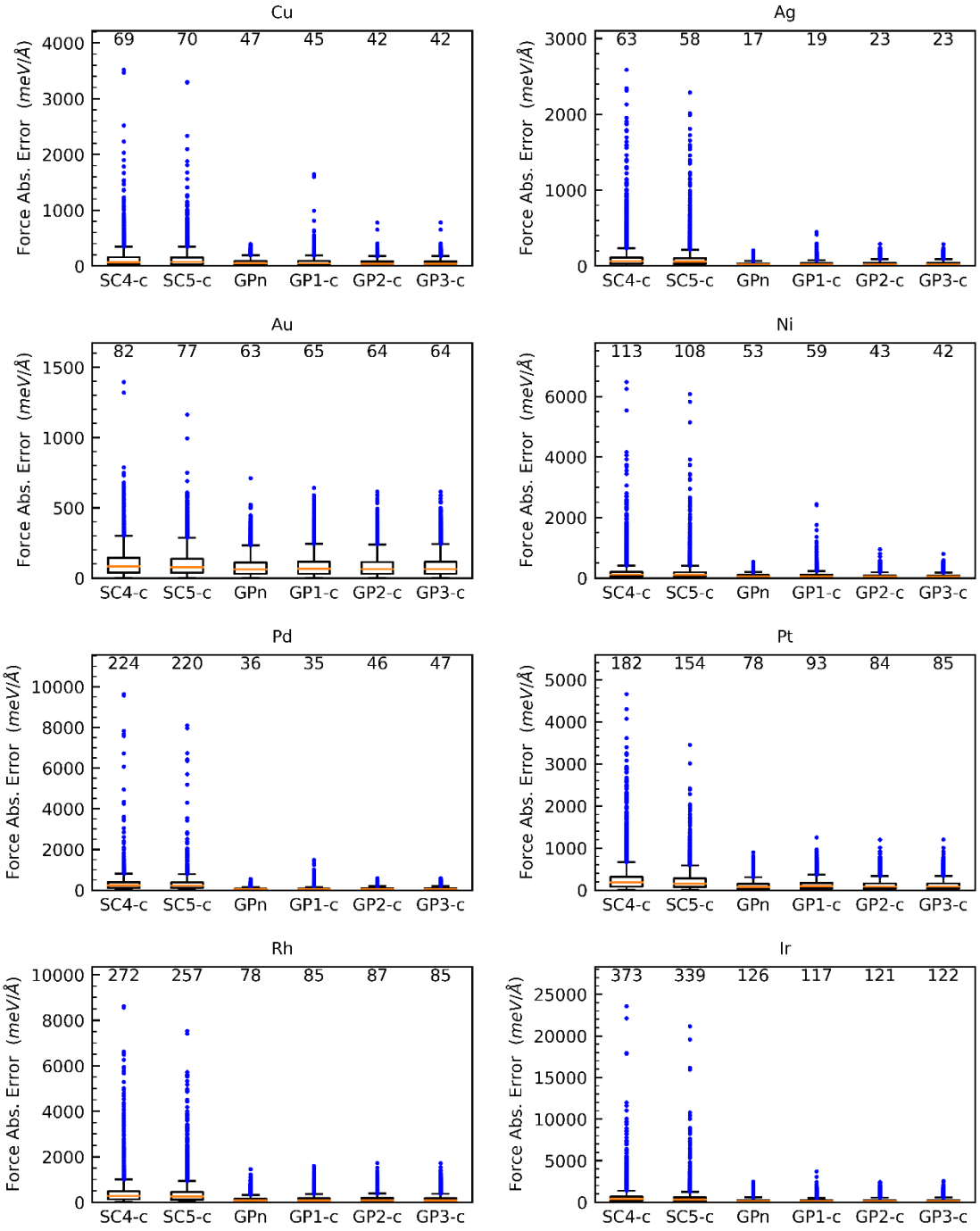


Figure 36. Box plots of absolute errors on the validation components of the force. The orange lines are the median, and the values of the medians are shown on top. The boxes show the interquartile range (IQR), and the whiskers are at 1.5 of the IQR, the blue points are the outliers beyond 1.5 of the IQR.

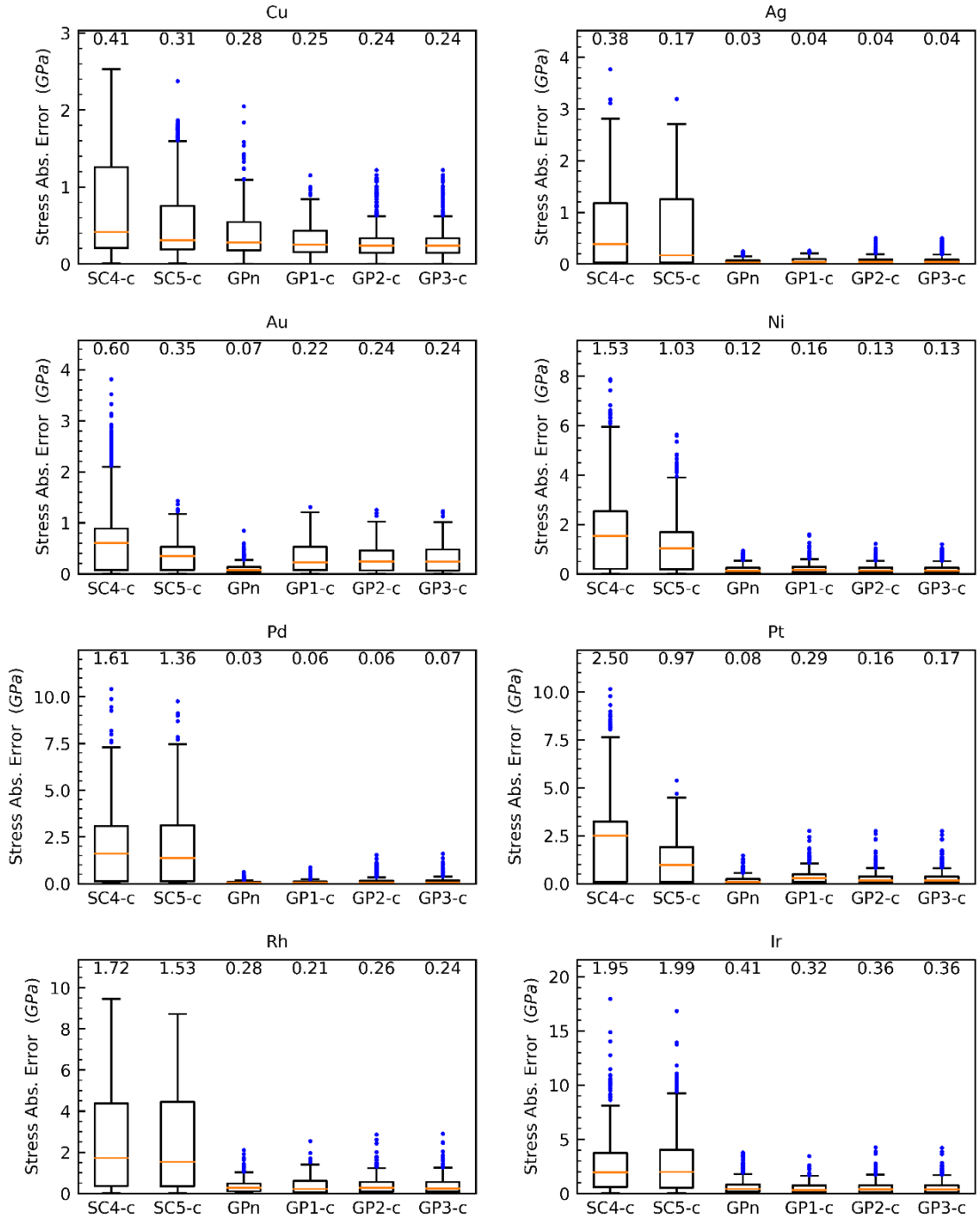


Figure 37. Box plots of absolute errors on the validation components of the virial stress tensor. The orange lines are the median, and the values of the medians are shown on top. The boxes show the interquartile range (IQR), and the whiskers are at 1.5 of the IQR, the blue points are the outliers beyond 1.5 of the IQR.

3.3.2 New functional forms identified using POET.

One of the advantages to the symbolic regression approach is that searches for new models can be "seeded" with models that are known to perform well and/or have foundations in physical principles. For the generation of the GPn models, POET was seeded with three different functional forms: GP1, GP2, GP3 [89], or Sutton Chen. A summary of which seeding method ultimately led to the model that was selected as "GPn" is provided in Table 22. For five of the eight elements (Ag, Au, Ni, Pd, and Ir), the "GPn" model came from a run seeded with the Sutton Chen model, demonstrating the benefits of starting from a physically derived functional form. The GPn forms for Ni and Pd inherited an embedding function that resembles the embedding function of the Sutton Chen model (Table 22), where the density is exponentiated to a fraction. The pair interaction term of Ni retained a similar form to the original Sutton Chen seed with an additional constant, but the pair term for Pd is different. While the functional form of the density of Pd remained the same as the Sutton Chen seed, the density of Ni was updated by POET using the natural exponential function with a power of e^{-ax}. Interestingly, the embedding function of the GPn models for Au and Ir have an embedding form like the one of GP1, which is a constant divided by the density. The pair interaction term of Au retained a similar form to the original Sutton Chen seed, and POET added a constant. The pair interaction term of Ag is like the one of Au, but the embedding component has a new form: a constant to the power of a summation over the interatomic distances with neighbor atoms. These observations suggest that POET successfully extracted information from the Sutton Chen seed, and combined it with new functional forms to find better models.

Table 22. POET GPn models (see appendix for full numerical precision).

Element	Seed	POET GPn model
Cu	GP1	$64.155\Sigma\left(\left(\frac{1}{r^{4.461}}-0.195^{r}\right)f(r)\right)+\frac{29.764}{\Sigma(f(r))}$
Ag	SC	$\Sigma\left(\left(\frac{424.997}{r^{7.322}}-0.011\right)f(r)\right)+8.173\times0.702^{\Sigma\left(r^{1-r}f(r)\right)}$
Au	SC	$\Sigma\left(\left(\frac{8167.212}{r^{10.860}}-0.011\right)f(r)\right)+\frac{0.054}{\Sigma\left(r^{-5.377}f(r)\right)}$
Ni	SC	$\Sigma\left(\left(\frac{43.845}{r^{3.698}}-0.148\right)f_2(r)\right)-62.560\left(\Sigma\left(re^{-1.885r}f_2(r)\right)\right)^{0.870}$
Pd	SC	$42.613\Sigma\left(r^{2.136r}f(r)\right)-42.613\left(\Sigma\left(r^{-4.790}f(r)\right)\right)^{0.071}$
Pt	GP2	$\Sigma\left(\left(10.892r^{5.037-3.650r}-0.037\right)f_2(r)\right)+12.720\times0.215^{\Sigma\left(3.649r^{-r}f_2(r)\right)}$
Rh	No seed	$\Sigma\left(\left(-89.551\times0.264^{r}-0.317+\frac{98.908}{r^{3.580}}\right)f(r)\right)+\frac{0.083}{\Sigma\left(0.136^{r}f(r)\right)}$
Ir	SC	$\Sigma\left(28.226r^{-1.942^{r}}f(r)\right)+\frac{78.243}{\Sigma(f(r))}$

In Table 22, the smoothing function f_2 is defined as:

$$f_2(r)=6x^5-15x^4+10x^3, x=(r_{out}-r)/(r_{out}-r_{in}) \tag{3.7}$$

The POET run that found the GPn model of Cu was seeded with GP1. The embedding function of GPn of Cu resembles the one of GP1, but the pairwise terms are different. The POET run that found the GPn interatomic potential of Pt was seeded with GP2. GPn for Pt has a pairwise term with a form like r^{a-br}, which is present in GP2, and this form is not present in other GPn potentials; suggesting that POET used the seed to find the GPn model for Pt. POET did not use a seed to find the GPn models for Rh. Interestingly, the embedding function of Rh like the one of GP1, which may be preferred by the algorithm because it has a small number of nodes.

We tested the suitability of POET for developing new simple, fast, and accurate functional forms for Cu, Ag, Au, Ni, Pd, Rh, and Ir. We identified new functions that have a lower error on the validation energies, forces and stresses than SC4-c, SC5-c, GP1-c, GP2-c, and GP3-c on all the elements except Rh, for which GP1-c has the best fitness (Figure 21). Interestingly, the POET run that found GPn for Cu was seeded with GP3, and in turn, the POET run that found GP3 was seeded with GP1 and GP2. These suggest that POET had more chances of getting closer to the global optimum and found GPn for Cu, which has the best fitness for Cu. Compared to GP3-c for Cu, the model GPn for Cu has MAEs that are lower by 1 meV/atom, 15 meV/Å, and 0.1 GPa. More importantly, the average error of GPn on all the transferability properties for Cu, Ag, Ni and Pd is comparable to or lower than the error of other models derived with genetic programming, at a comparable level of complexity (Figure 22).

The GPn models developed with POET perform better on elements close to Cu in the periodic table (Figure 38) than in elements far from Cu; a similar trend is observed for GP1-c, GP2-c, and GP3-c (Figure 39). The trends in SC4-c and SC5-c models are similar (Figure 38 and Figure 39), where their errors are comparable on Cu and elements on its group on the periodic table, and the worst relative performances are for Ir, which is furthest from Cu. Interestingly, the heatmaps for SC4-c and SC5-c similar (Figure 38 and Figure 39) show that these two functions do not perform well on Pd, which is surprising given that Cu is close to Pd on the periodic table. Even though the GPn functional forms have a better accuracy on validation data than SC4-c and SC5-c for all the elements considered, the performance of GPn models on Au, Pt, Rh and Ir is significantly worse than on the other elements, suggesting that the hypothesis space needs to be expanded (e.g. by adding bond-

angle terms[29-30]) to develop suitably accurate models for these elements. The addition of new descriptors should also help POET discover interatomic potential models for systems where covalent bonding is important.

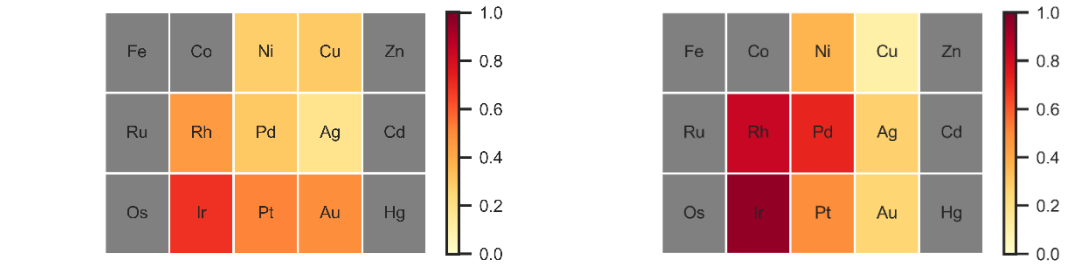


Figure 38. Average of normalized errors across validation properties for (a) GPn models, and for (b) SC4-c models. The validation metrics considered on this plot are: MAE of energies, MAE of forces, MAE of stresses, MAPE of C_{11}, C_{12} and C_{44}, MAPE of 7 phonon frequencies, absolute percent error of vacancy formation energy, absolute percent error of vacancy migration energy, absolute percent error of dumbbell formation energy, MAPE of 13 low-index surface energies, absolute percent error of intrinsic stacking fault energy, absolute percent error of unstable stacking fault energy, absolute percent error of hcp formation energy, absolute percent error of bcc formation energy, absolute percent error of fcc lattice parameter, and absolute percent error of bcc lattice parameter. The normalization was done using min-max scaling (x-min(x))/(max(x)-min(x))

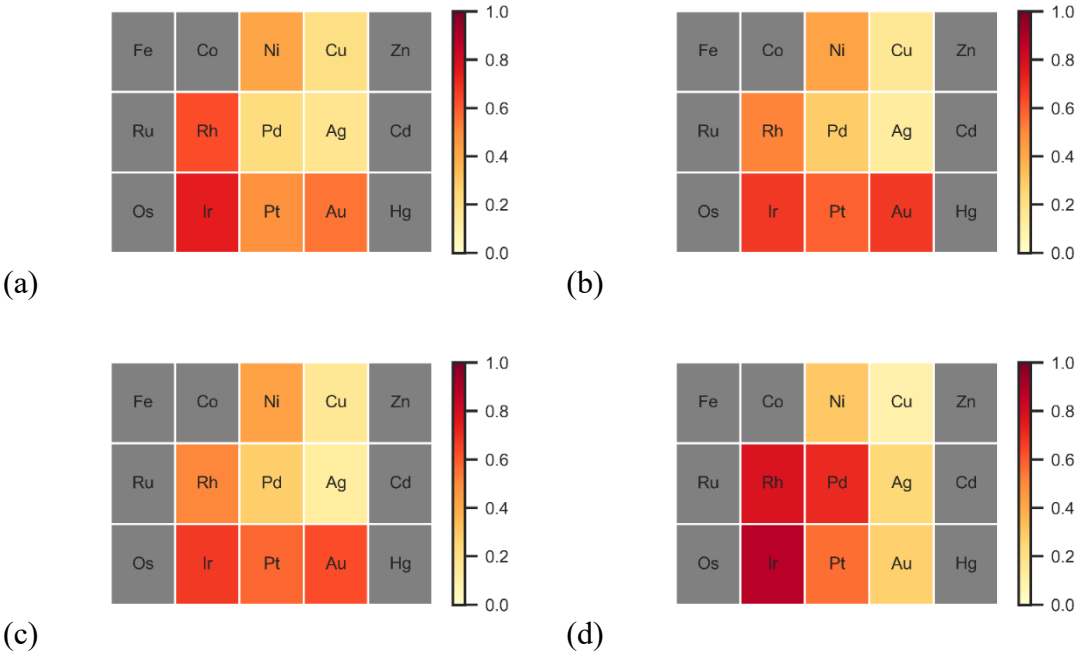


Figure 39. Average of normalized errors across validation properties for the models (a) GP1-c, (b) GP3-c, (c) GP2-c, and (d) SC5-c. The validation metrics considered on this plot are: MAE of energies, MAE of forces, MAE of stresses, MAPE of C_{11}, C_{12} and C_{44}, MAPE of 7 phonon frequencies, absolute percent error of vacancy formation energy, absolute percent error of vacancy migration energy, absolute percent error of dumbbell formation energy, MAPE of 13 low-index surface energies, absolute percent error of intrinsic stacking fault energy, absolute percent error of unstable stacking fault energy, absolute percent error of hcp formation energy, absolute percent error of bcc formation energy, absolute percent error of fcc lattice parameter, and absolute percent error of bcc lattice parameter. The normalization was done using min-max scaling (x-min(x))/(max(x)-min(x))

The Generalized Gradient Approximation (GGA) [129] has been reported to have problems in calculating the vacancy formation energy of some fcc metals. [170-171] Interestingly, the vacancy formation energies predicted by the GPn models, which were extrapolated from bulk DFT data, are closer to the experimental values than the DFT values are. For the vacancy formation energies computed with DFT and GPn, the average difference to the closest experimental value for GPn is 65 meV compared to 214 meV for DFT (Figure 40). The vacancy formation energies predicted by GPn for Cu, Ni, Pd, and Pt are within the

experimental range. For Ag, Au and Rh, the errors are less than 100 meV between the predictions of GPn and the closest experimental vacancy formation energy, and the error is 281 meV for Ir. O'Brien et al. observed a similar trend for the vacancy formation energies of Pt and Au many-body interatomic potentials. [172] They reported a general good agreement between the vacancy formation energies predicted by EAM potentials and experimental values when fitting the models with DFT data excluding atomic configurations with vacancies.

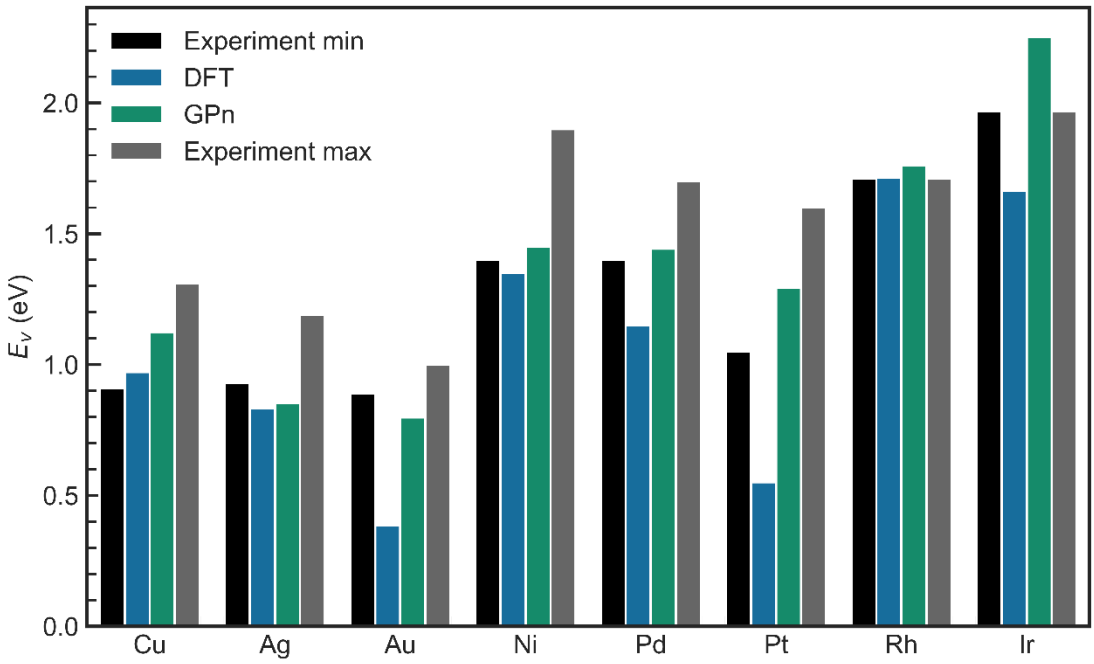


Figure 40. Vacancy formation energy (E_v) predicted by GPn models compared to DFT E_v, maximum experimental E_v, and minimum experimental E_v. See Table 23 for a list of references of the experimental values.

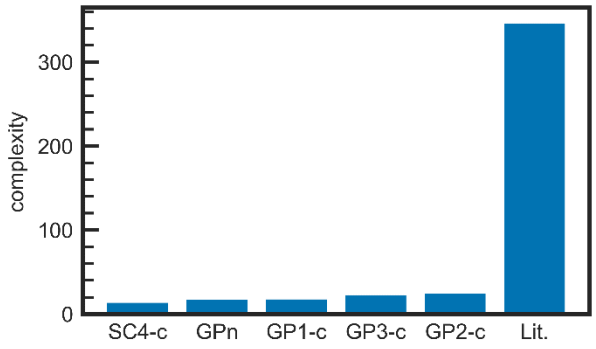
Table 23. Literature references of the experimental vacancy formation energies used for determining the maximum and minimum experimental vacancy formation energy for each element.

Element	References of the experimental E_v
Cu	173-180
Ag	173, 175, 178, 180-181
Au	173, 175, 180, 182
Ni	175, 183-184
Pd	143, 185-186
Pt	143, 173, 186-187
Rh	188
Ir	188

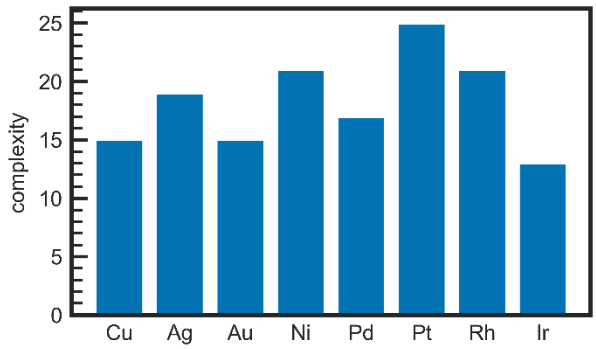
3.3.3 Assessing the tradeoff between accuracy and complexity: validating against literature EAM-type models

We compared the performance of the interatomic potential models developed with genetic programming against the EAM-type models in the NIST Interatomic Potentials Repository [37] for Cu, Ag, Au, Ni, Pd, Pt, Rh and Ir, and against the EAM models from Sheng et al[176]. To assess the accuracy and complexity tradeoff of the EAM-type interatomic potential models developed in this chapter, we generated a Pareto frontier for each element (Cu, Ag, Au, Ni, Pd, Pt, Rh and Ir) and for each of the 21 validation property listed in the caption of Figure 42, giving a total of 168 Pareto frontiers. In each Pareto frontier, the y-axis was the error on a validation property, and the x-axis was the model complexity (i.e., number of nodes). The models on the Pareto set are optimal in the multi-objective target of achieving low complexity and low error on validation properties; no model is less complex and has less error than a model in the Pareto frontier. These Pareto frontiers are relevant for two main reasons: (1) they indicate that interatomic potential models developed with genetic programming usually have less error than the simple Sutton Chen models, and (2) they imply that the models developed from genetic programming have less error than

several of the models from the literature (which are on average more complex). As an example of how the Pareto frontiers work, the models GP1-c, GP2-c, and GP3-c are on the Pareto frontier of vacancy migration energy for Ni before GPn. The models GP3-c and GP2-c drop from the Pareto frontier after adding GPn for Ni because it is more accurate and less complex, with 21 nodes (Figure 42 (a) and Figure 41).



a)

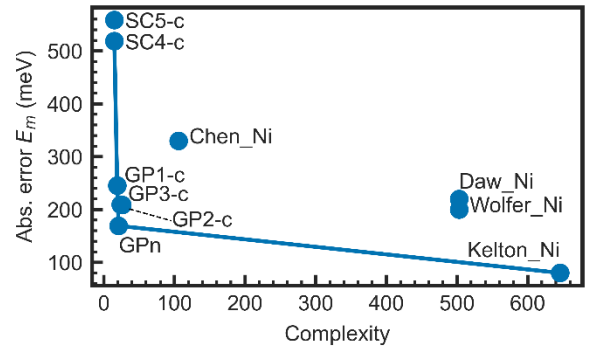


b)

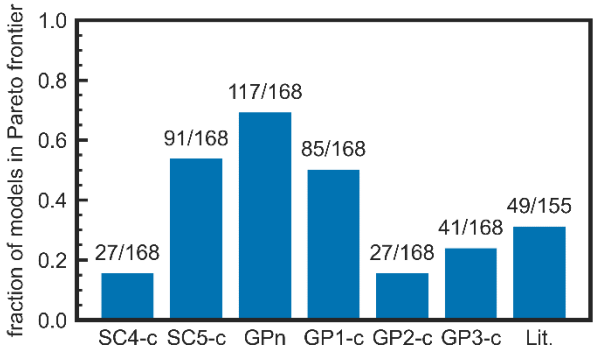
Figure 41. (a) Complexity (i.e., number of nodes) of each of the model types. The number of nodes of SC4-c and SC5-c are the same. The number of nodes of the models are 15, 19, 24, and 26 nodes for SC4-c, GP1-c, GP3-c, and GP2-c, respectively. The average number of nodes of GPn models is 18, and the average for literature models is 348. (b) Complexity (i.e., number of nodes) of the GPn models for each element.

The models GPn, GP1-c, GP2-c, and GP3-c have a good balance of accuracy and complexity because these models belong to the Pareto frontiers a similar number of times when competing against SC4-c, SC5-c, and literature EAM-type interatomic potential

models. The simplicity of GP1-c, with 19 nodes, makes it competitive on the tradeoff of error and complexity because it is just 4 nodes more complex than Sutton Chen and frequently has less error. Therefore, GP1-c belongs to 51% of the Pareto frontiers, compared to 54% for SC5-c. If GP1-c had more error than SC5-c on many properties for many elements, then it would not be in the Pareto frontier because SC5-c would be both more accurate and less complex.



a)



b)

Figure 42. (a) Pareto frontier of EAM-type interatomic potential models for Ni considering the absolute error on the vacancy migration energy and the number of nodes (complexity). No model has less error and is simpler than a model in the Pareto frontier. The models SC4-c, GP1-c, GPn and Kelton_Ni belong to the frontier. The references are: Chen_Ni [189], Daw_Ni [144], Wolfer_Ni [143], and Kelton_Ni [190]. (b) Number of times that an EAM-type model belongs to the Pareto frontier divided by the number of times that the model has validation values available across the elements and properties. The metrics considered are the validation MAE on energy, force, and stress, MAPE on elastic constants, MAPE on 13-low index surface energies, absolute error on vacancy formation energy, absolute error on vacancy migration energy, absolute error on dumbbell formation energy, absolute error on intrinsic stacking fault energy, absolute error on unstable stacking fault energy, absolute error on hcp formation energy, absolute error on bcc formation energy, absolute error on fcc lattice parameter, absolute error on bcc lattice parameter, and absolute error on each high-symmetry phonon frequency: $\nu_L(X)$, $\nu_T(X)$, $\nu_L(L)$, $\nu_T(L)$, $\nu_L(K)$, $\nu_{T1}(K)$, and $\nu_{T2}(K)$.

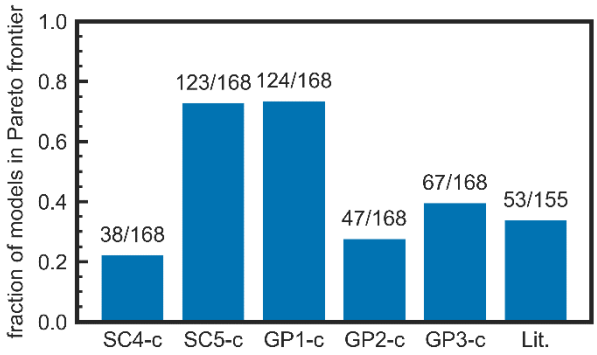


Figure 43. Number of times that an EAM-type model belongs to the Pareto frontier divided by the number of times that the model has validation values available across the elements and properties, excluding GPn models to analyze the transferability of GP1-c, GP2-c, and GP3-c. The metrics considered are the validation MAE on energy, force, and stress, MAPE on elastic constants, MAPE on 13-low index surface energies, absolute error on vacancy formation energy, absolute error on vacancy migration energy, absolute error on dumbbell formation energy, absolute error on intrinsic stacking fault energy, absolute error on unstable stacking fault energy, absolute error on hcp formation energy, absolute error on bcc formation energy, absolute error on fcc lattice parameter, absolute error on bcc lattice parameter, and absolute error on each high-symmetry phonon frequency: $\nu_L(X)$, $\nu_T(X)$, $\nu_L(L)$, $\nu_T(L)$, $\nu_L(K)$, $\nu_{T1}(K)$, and $\nu_{T2}(K)$.

The new functional forms developed in this thesis, GPn, reduce the number of times that SC4-c, SC5-c, GP1-c, GP3-c, GP2-c, and literature models belong to the Pareto frontiers by 7, 19, 23, 15, 12, and 3 percent, respectively (comparing Figure 42 (b) and Figure 43). The average complexity of GPn interatomic potential models is 18 nodes. Their simplicity and accuracy cause an improvement on the Pareto frontiers. GPn has less error on vacancy migration energy than 75% of the literature models (3 out of 4 in Figure 42 a). Performing a similar analysis on each Pareto frontier for all validation properties and all the elements, we find that remarkably the errors of the models GP1-c, GP2-c, GP3-c, and GPn, are lower than 46, 51, 52, and 53 percent of the EAM-type literature models, respectively, and they

are on average more than 10 times simpler. In contrast, the errors of the models SC4-c and SC5-c are more accurate than 34 and 35% of EAM-type literature models, respectively.

3.4 Conclusion

In conclusion, we demonstrated that the functional forms of interatomic potential models developed for Cu with POET have good transferability to Ag, Ni, and Pd, which are close to Cu on the periodic table, but their performance is not as good on other elements further from Cu. We report a similar trend for new GPn functional forms developed with POET in this chapter. The good performance on elemental systems like Cu may suggest that the functional forms of GP1, GP2, and GP3 encoded physical information that allows them to transfer well to these elemental systems. Even though POET was able to identify functional forms that outperform the Sutton Chen functions across Cu, Ag, Au, Ni, Pd, Pt, Rh, and Ir, the accuracy on Au, Pt, Rh, and Ir is not high. To overcome this limitation, the set of mathematical terms and descriptors of the atomic environment from which POET builds models should be expanded, or the algorithm should focus on identifying more complex functional forms. The focus on expanding the hypothesis space should be preferred because it would allow maintaining simple functional forms. Finally, we showed that the models developed with POET show a good balance of accuracy and complexity, with a competitive percentage belonging to the Pareto frontier of absolute error and number of nodes across elements and validation properties when competing against Sutton Chen and other EAM-type models from the literature.

4 Developing a database of atomically precise nanoclusters

4.1 Background and summary

The physical and chemical properties of atomically precise nanoclusters are different from the bulk properties, including discrete energy levels,[191-192] nonlinear optical properties,[193] magnetism,[194] high catalytic activity,[195] multiple absorption bands,[196] and enhanced photoluminescence[197-200]. The high number of atoms with low coordination number on the surface of these clusters and their small size give rise to their novel properties, [201-202] which are a function of the size and shape of the cluster, facilitating "bottom-up" materials design. [201-202] Researcher have made significant progress in computational methods to predict the properties of nanoclusters in the past few decades, however the determination of the atomic structure of the cluster remains challenging. [74, 76, 107, 203-204] Researchers have developed methods to search for the ground state atomic structure by sampling the potential energy surface (PES) of the cluster. The approaches include genetic algorithms, [204-206] particle swarm optimization,[207] Bayesian optimization[208] and basin-hopping[209] methods. The general idea behind these techniques is to find low energy clusters by searching atomic coordinates that minimize the energy of the cluster; the methods usually compute the energy of many structures.

The search for atomic coordinates that minimize the energy is a computationally expensive task due to the cost of computing the energy of the system and the large size of the search space. It is estimated that the number of potential energy minima increases exponentially with the size of the system and the number of dimensions to optimize grows as O(N) with the number of atoms in the cluster, with a prefactor of three.[210-211] At the atomic scale,

quantum and finite-size effects are significant, and the relative energies of clusters can be accurately determined using *ab initio* methods. Current nanocluster datasets are either unavailable to the public, limited in scope, or primarily use lower levels of theory like interatomic potential models and tight binding models. There is a need and interest in predicting the structures and properties of atomically precise nanoclusters at higher levels of theory and making the data publicly available.

We addressed this challenge by creating a publicly accessible database of more than 50,000 atomically precise nanoclusters that have low energy, called the Quantum Cluster Database. We performed high-throughput density functional theory (DFT) calculations with a genetic algorithm and identified clusters of up to 55 atoms for 55 different elements in the periodic table. The 55 elements encompass different regions of the periodic table, including alkali and alkaline earth metals, transition metals, post transition metals, metalloids, and non-metals. To the best of our knowledge, this database constitutes the most extensive collection of computed bare cluster structures at the DFT level of theory. The data set can be used to guide experimental synthesis of predicted nanoclusters, to computationally screen for clusters suitable for a variety of applications, or to train machine learning models. Since the structural energies were obtained using a consistent computational method, the data also serves as a direct source for comparative benchmark studies of different DFT or other electronic structure techniques within the context of atomic cluster modelling. All atomic structures and their calculated properties are openly distributed, enabling researchers across the world to access it for free and use it for further analysis.

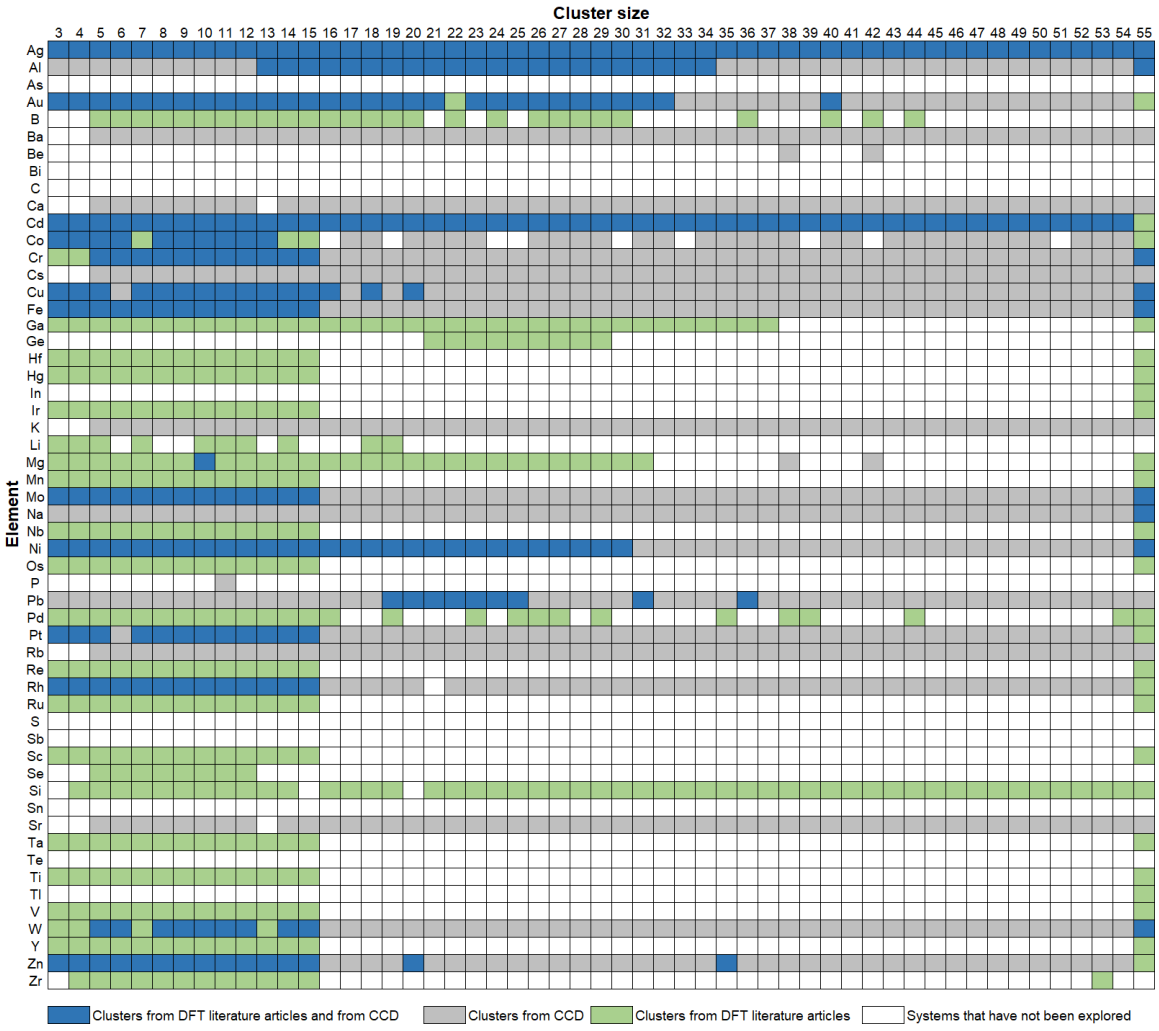


Figure 44. A summary of previous studies of elemental clusters in terms of exploring their atomic structures. We have considered literature that used DFT to find atomic structures as well as systems covered in the Cambridge Cluster Database that used empirical potentials. We have considered 55 different elements across the periodic table with size regimes from 3-55 atoms in the cluster. Note: even when the cluster of a particular element and size has been explored in the literature, we may have discovered new clusters for that element and size.

4.2 Methods

4.2.1 Identifying low energy clusters using a Genetic Algorithm

A genetic algorithm belongs to the category of evolutionary algorithms, and it is draws inspiration from Darwin's theory of natural selection.[98-99] Genetic algorithms probabilistically explore large spaces and perform a global search applying the operations of mutation and crossover to a set of individuals; the individuals are clusters in our case and the set of individuals is generally called the "population". In this chapter, we also use a "seed" operation, described in more detail below. The mutation, crossover and seed operations generate new clusters from the parent clusters, and the best members from the parents and the children are kept as the new parents for the next iteration of the search until the population stagnates for a certain number of iterations. Genetic algorithms have been applied to search globally stable configurations for many types of structures, including elemental metal clusters,[212-215] alloys clusters,[216-217] surface supported nanoparticles,[218-221] and bulk crystalline materials[222-223]. We implemented our own genetic algorithm code based on the Birmingham Parallel Genetic Algorithm with some modifications.[224-225] We maintain a pool of a fixed size, which contains the lowest energy clusters from the set of all structures. The initial members of the pool can be either randomly generated structures or custom seed structures. The algorithm generates random structures by randomly distributing atoms in a cubic box with a side length of $2r\sqrt[3]{N}$ where r is the atomic radius of corresponding element and N is the number of atoms. The atomic radii are retrieved from an internal list compiled from elemental DFT calculations. Clusters start filling the pool, and when it is reaches the maximum size, the crossover, mutation, and seed operations are applied to generate new structures from parents selected from the pool. For

the crossover operation, a pair of parent clusters are picked from the pool based on a probability that is proportional to a numerical value called the "selectability". The selectability of the i^{th} cluster is:

$$S_i = f_i \times v_i \tag{4.1}$$

where f_i represents the fitness of the i^{th} cluster and v_i is a penalty term.[226] The fitness is rewards clusters with low energies and is calculated using a function of a normalized total energy:

$$f_i = \frac{1}{2}[1 - \tan(2\rho_i - 1)] \tag{4.2}$$

and the normalized energy:

$$\rho_i = \frac{E_i - E_{\min}}{E_{\max} - E_{\min}} \tag{4.3}$$

where E_{min} and E_{max} are the lowest and highest total energies of pool clusters.[224] The next term in the selectability (equation (4.1)) is the penalty. The penalty (equation (4.4)) prevents oversampling a particular cluster, and it avoids overcrowding child generation with descendants of same parent.

$$v_i = \frac{1}{1 + \sqrt{\eta_i}} \tag{4.4}$$

where η_i is the selection frequency of the i^{th} cluster during the entire genetic algorithm run.[226] It has been shown that incorporating this penalty term can improve both the convergence speed and the success rate of a genetic algorithm for identifying the global optimum.[226] Once a pair of parents have been selected as parents, the cut-and-splice method from Deaven and Ho is applied to generate a child cluster.[227] The child can inherit either

half of the total atoms or a random share from each parent. For the mutation operation, a child cluster is generated by randomly displacing and rotating 20% of the atoms of the parent cluster. Crossover is used as the primary genetic operation and mutation is used to keep the ratio of pool clusters that are created by mutation at 20%. As for seeding operation, it creates new clusters from seed structures which have either different sizes or different element species to the current system. In the former case, atoms are either added or subtracted from seed structures to reach the desired size, whereas in the latter case, atoms are swapped to the target element and atomic coordinates are scaled in proportion to the ratio of atomic radii. If seed structures are provided, the initial members of the pool can also be filled by seeding operation. In this work, the initial population was not seeded is since all calculations started from scratch without prior knowledge of high-quality clusters. As a last step, a structure check is applied to detect overlap in newly created structures to prevent unrealistic configuration. If overlap exists, the corresponding child clusters are discarded, and new clusters are generated using the same type of operation until the overlap problem is not present.

The atomic coordinates of the children clusters are then relaxed with DFT using the Vienna Ab initio Simulation Package[128] (VASP). The side length of simulation box ensures that periodic images are separated by a distance of at least three times the nearest neighbor distance. The child clusters are introduced to the pool if they have lower energy than the pool clusters of high energy and if they are not structurally equivalent to other pool cluster.[228] To determine whether two clusters are structurally equivalent, a geometric similarity is used; two clusters are not equivalent if they have a similarity score greater than 0.3. When a child cluster is equivalent one or more pool clusters and lower in energy, it

will replace either the most similar pool cluster if score is smaller than 0.1 or the highest energy pool clusters if all scores are larger than 0.1. The employment of a similarity check guarantees structural diversity of the pool and avoids multiple occurrences of same configuration in the pool.

The pool size is another factor that affects the performance of a genetic algorithm. A smaller pool size is shown to increase convergence speed, whereas a larger one can improve success rate of identifying the global minimum.[224, 226] We used a population size of 10 for all genetic algorithm runs as a balance between global search and convergence speed. A genetic algorithm run was stopped when the total number of generated clusters exceeded 1000. A schematic diagram summarizes this workflow is shown in Figure 45.

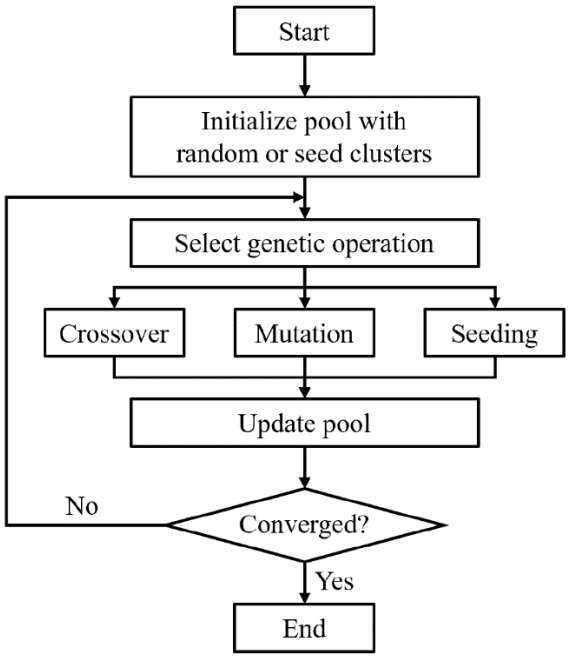


Figure 45. Schematic workflow of the genetic algorithm used for the Quantum Cluster Database

4.2.2 Correlations between elements in the Quantum Cluster Database

The clusters of some elements in the Quantum Cluster Database are correlated (or negatively correlated) with other elements (Figure 47). Knowing the correlations is valuable because it can be used to identify new low-energy clusters or to perform exploratory analyses. These are steps that we performed to generate the correlations (Figure 47):

1. Searched low-energy clusters using the genetic algorithm for the sizes 5, 10, 15, and 20 for the elements Al, Be Li, Mg, Na, Si, Ta and Ti. These elements cover different areas of the periodic table and they are computationally inexpensive.
2. Calculated the energies of all the elements in the Quantum Cluster Database using clusters from step 1 as templates and scaling the interatomic distances with the hard sphere radii.
3. Identified a few elements that are most different from all other elements. We used a linear model of the energies of one element of step 2 as a function of the energies of all other elements from step 2. The elements with worst fits are the ones that are most different to the rest. We selected the 13 most different elements: B, Ba, Be, Ca, Cr, Cs, K, Li, Mg, Na, Rb, Sr and Zn. This step is important for computing the correlations because it accounts for different types of structures across elements. As a counter example, if we used only one type of structure (like a structure characteristic of clusters that correspond to transition bulk fcc metals), we would get the correlations of between different elements in the space covered by that

specific structure (transition fcc elements in this example), and we would not be able to accurately identify correlations across elements.

4. Found low-energy clusters of the 13 elements identified in step 2 for the sizes 10, 15, 20, 25 and 30 (Figure 46), using the genetic algorithm.
5. Used the low-energy clusters from step 3 as templates to compute the energies of all the elements in the Quantum Cluster Database, scaling the interatomic distances.
6. Computed the table of Pearson correlations using the energies from step 4 (Figure 46).

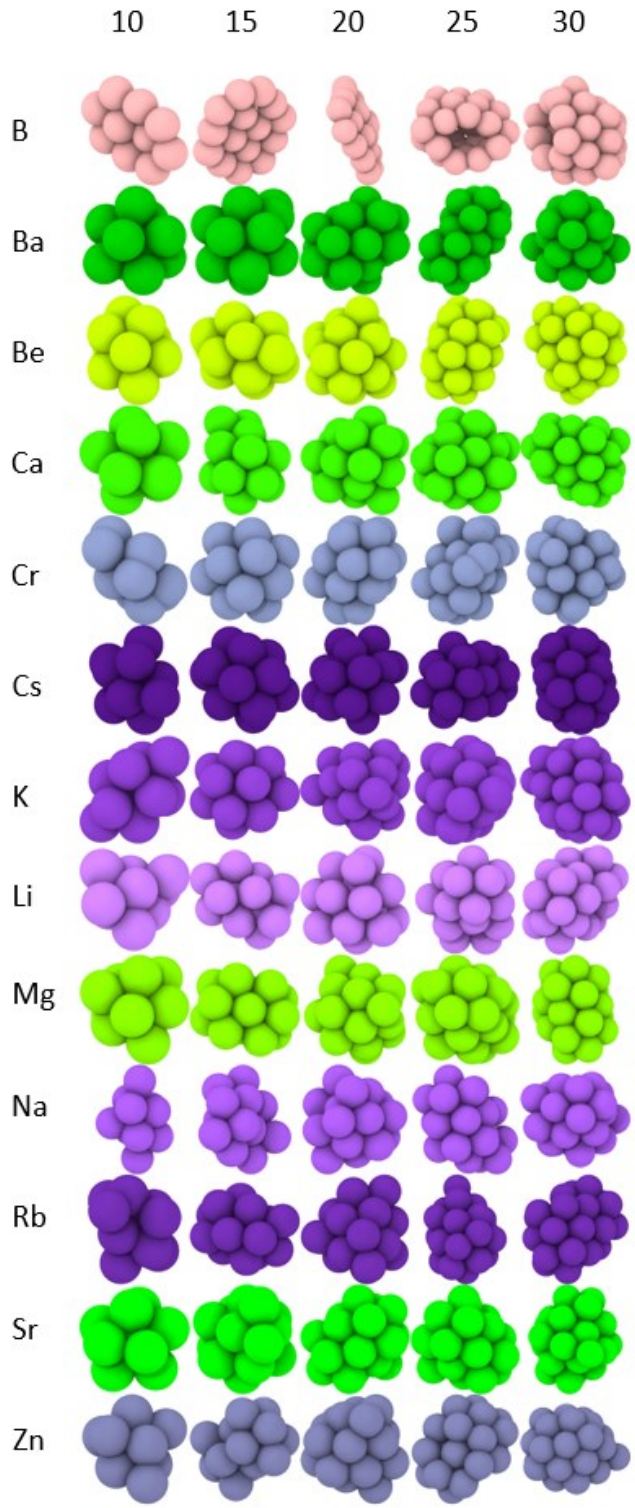


Figure 46. Template clusters used for identifying low-energy clusters using correlations.

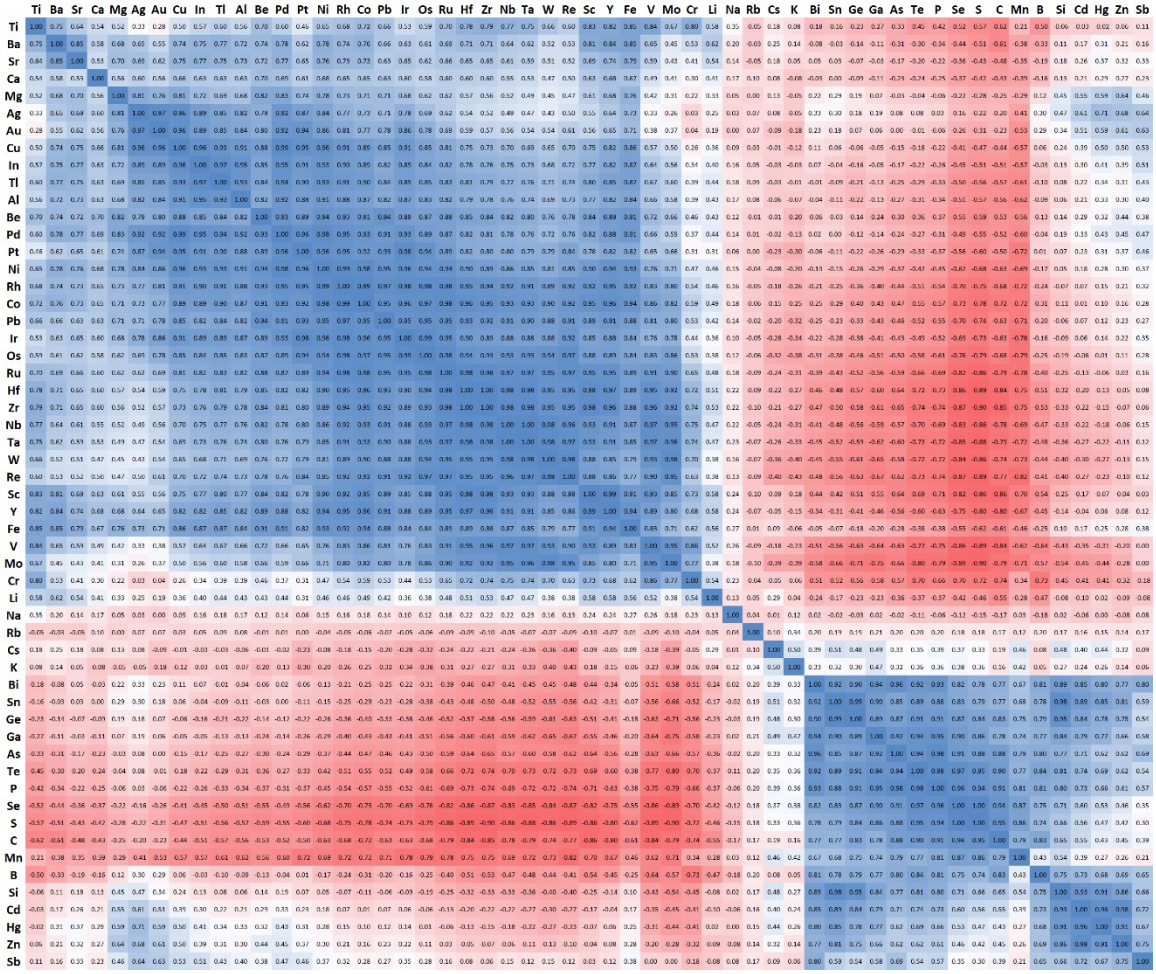


Figure 47. Pearson correlation coefficients between elements in the Quantum Cluster Database sorted to facilitate the identification of trends.

As an example, the correlation coefficients may be used to find new low-energy clusters. For example, Ti and Zr have a high correlation of 0.79. The genetic algorithm search using DFT for Zr clusters is more expensive than the search for Ti clusters. One can search clusters for Ti using the genetic algorithm and then use the low-energy Ti clusters as templates for finding low-energy Zr clusters, without the need for running the genetic algorithm on Zr. Alternatively, it would be possible to use the Ti template clusters as a seed to accelerate a genetic algorithm run for low-energy Zr clusters.

4.2.3 DFT calculations

All local energy minimizations were carried out using the Vienna ab initio Simulation Package [128] (VASP) by means of the Perdew-Burke-Ernzerhof [129] (PBE) generalized gradient approximation (GGA). The projected-augmented wave [130] (PAW)

pseudopotentials used in VASP are shown in Table-Appx. 5. All calculations were run at the gamma point with spin polarization. Methfessel Paxton smearing with = 0.01 eV was implemented to improve SCF convergence.

4.2.4 Workflow

The workflow for generating the data for the Quantum Cluster Database is shown in Figure 48. The first step in the process was to generate the clusters and import them from the literature whenever possible. We have discovered new clusters by using the genetic algorithm described in this work. To create a comprehensive and consistent database, we imported the clusters from the literature, and computed their properties using DFT with our settings.

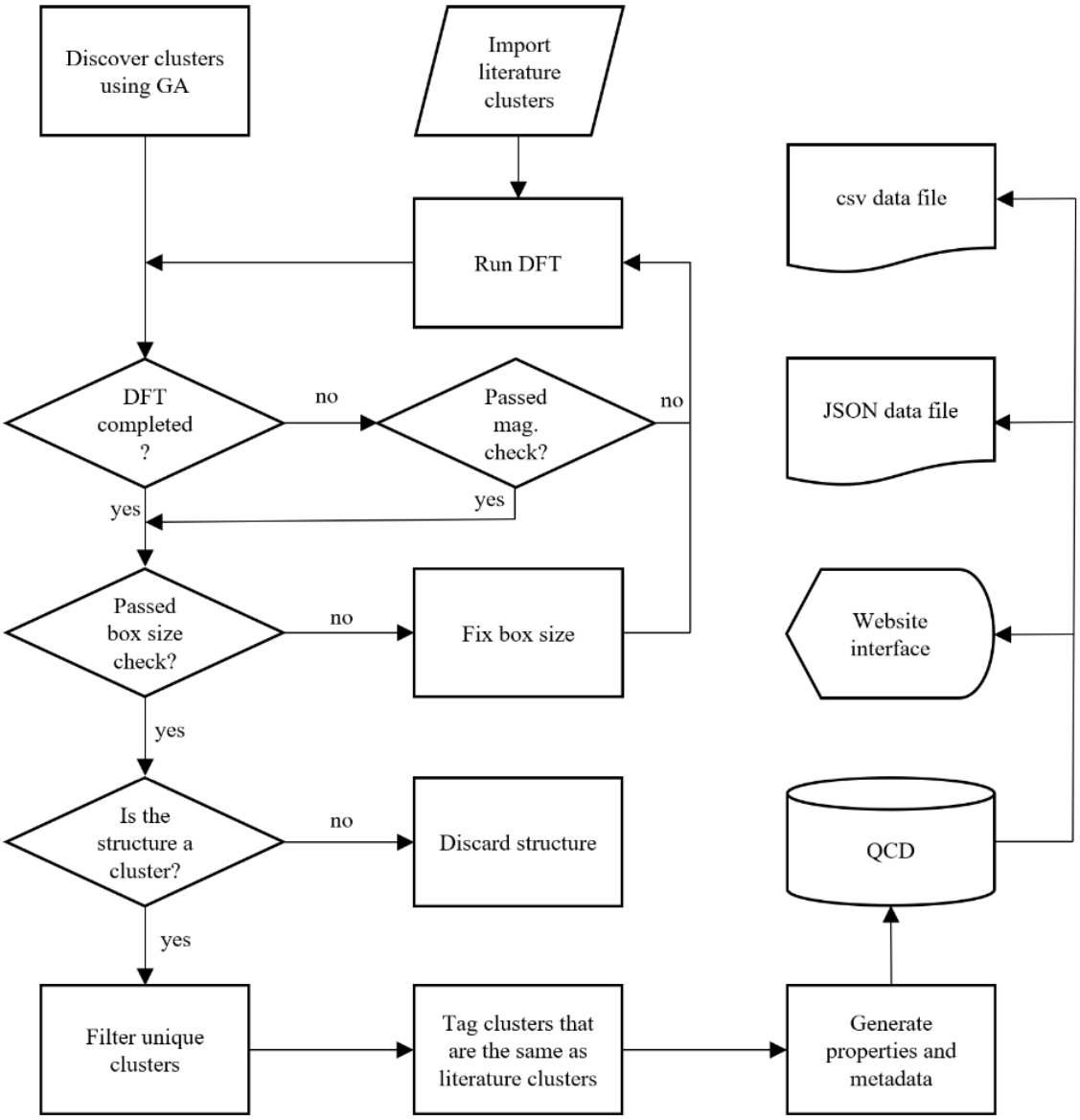


Figure 48. Workflow for generating the Quantum Cluster Database.

The next step was to check that the DFT calculations completed successfully. In this step, we verify that the desired settings were used in the INCAR file (like default cutoff energy, precision, and appropriate spin flags and initial magnetic moments). We then check the OUTCAR file to verify that the settings printed in this file match the settings from the INCAR file, and that the structure printed in the OUTCAR file matches the structure from

the POSCAR file. We also check that the force converged within 0.15 eV/angstrom, and that the OUTCAR finished. We further check the OUTCAR file to confirm that the pseudopotential name matches the name in the POTCAR file, and that the correct pseudopotential was used.

After checking that a DFT calculation completed successfully, we check whether the calculation needs to be re-run. The calculation does not need to be re-run if the only issue with the DFT the calculation is that the forces did not converge and the structure ran with various initial magnetic moments, from which at least one completed successfully. Otherwise, the DFT calculation is computed again.

The "box size" check verifies that the minimum periodic distance between images is large enough. For this, we have two types of checks. The first check considers elements in the Groups 1A and 2A on the periodic table, for which the minimum distance between periodic images must be greater than 3.5 times the distance between nearest neighbors in the ground state structure taken from the Materials Project. The second check considers all other elements, for which the minimum distance between periodic images must be greater than 10 angstroms. If the minimum distance between periodic images is small, then we increase the supercell size and then ran the DFT calculation again.

Following the box size check, we check that the structure corresponds to a cluster. We simulate the structures under periodicity conditions, for which we use a large supercell size that creates vacuum around a group of atoms to generate a cluster. However, the atoms sometimes assume periodic configurations that correspond to nanowires, or or slabs. We filter out these types of structures by discarding clusters that have a minimum distance between periodic images smaller than 1.5 times the nearest neighbor distance. Other type

of erroneous structure that is filtered out by the "Is structure a cluster?" check are discontiguous clusters. An example of a discontiguous cluster is a structure that separated into two clusters within the same simulation box; these types of structures are discarded. This filter uses a distance of 1.5 times the nearest neighbor distance to check whether the structure is contiguous.

The Quantum Cluster Database contains only unique clusters for a given element and size. The step that filters duplicate clusters uses the similarity calculator described previously, with a similarity threshold of 0.3. The next step in the workflow considers that the genetic algorithm or the correlations method may find clusters that are equivalent to clusters from the literature, and such clusters are tagged as literature clusters in that case.

Then, the properties and metadata described in the Data Records section are generated for each cluster, and they are stored in a PostgreSQL database. Finally, the data is displayed in the Quantum Cluster Database website and output as a JSON file and a .csv file.

4.3 Data records

The files from the density functional theory calculations of the more than 50,000 clusters are publicly available through the Materials Cloud repository, and a web interface to visualize the structures, the correlations, and the properties can be accessed through the website of the Mueller Research Group, from where the data can be downloaded in as a JSON file or as a comma-delimited file.

Table 24. Keys, types of data, and description of the QCD data in the JSON file and .csv format

Key	**Datatype**	**Description**
cluster_id	string	ID of the cluster in QCD
element_symbol	string	symbol of the element of the cluster
n_atoms	number	number of atoms in the cluster
n_val_electrons	number	number of valence electrons corresponding to the pseudopotential
energy_dft	number	energy in eV
energy_relative	number	energy in eV above the lowest energy structure of the same element and size
energy_n_minus_one	number	formation energy in eV relative to the lowest energy structure of the same element but of size N-1
energy_n_plus_one	number	formation energy in eV relative to the lowest energy structure of the same element but of size N+1
homo_lumo_gap	number	HOMO-LUMO Gap in eV
magnetic_moment	number	Magnetic moment of the cluster in units of Bohr magneton (μ_B)
similar_structures	list	space delimited list of cluster_id of clusters within QCD that are similar to this cluster
references	list	space delimited list of literature references
structure_xyz	string	structure represented in XYZ format [(a)]
structure_poscar_format	string	structure represented in POSCAR format [(a)]

Notes: (a) semicolons are used instead of line breaks.

4.3.1 File format

The data is available for download as a JSON file and as a .csv file. Both can be downloaded from the Mueller Research Group website. The first level of the JSON file contains an arbitrary index for every cluster, the next level contains the cluster_id described in Table 24, and the next level contains the other keys described in Table 24, with the

corresponding values. The columns of the .csv file correspond to the keys described in Table 24. The VASP DFT calculation files for each cluster are available in the Materials Cloud repository in the form of text files from the inputs and outputs of VASP.

4.3.2 Properties

For each cluster of a given number of atoms N and element type k, the database contains the energy relative to the lowest energy structure of size N and species k, the formation energy with respect to the stable cluster of size N-1 of species k (equation (4.5)), the formation energy with respect to the N+1 stable cluster of the same species (equation (4.6)), the HOMO-LUMO gap, the number of valence electrons considered by DFT, the magnetic moment, a list of similar structures within the Quantum Cluster Database, a list of literature references for the cluster (downloadable in .bib format), the coordinates (downloadable in XYZ format), and an interactive visualization of the cluster.

$$E_{f,N,N-1} = E_N - E_{N-1} - E_{atom} \tag{4.5}$$

$$E_{f,N,N+1} = E_N + E_{atom} - E_{N+1} \tag{4.6}$$

where E_N is the energy of the cluster of size N, E_{N-1} is the energy of the cluster of size N-1, E_{N+1} is the energy of the cluster of size N+1, and E_{atom} is the energy of an atom.

4.4 Technical validation

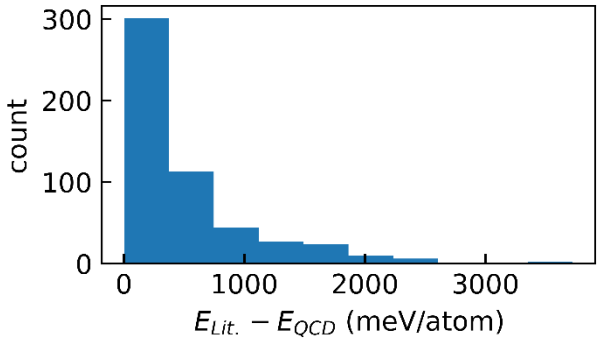


Figure 49. Count of the difference between the energy of the lowest-energy clusters found in the literature (minus 1 meV/atom to account for DFT precision) minus the energy of lowest-energy clusters discovered in this work. The Quantum Cluster Database work discovered 501 lowest-energy clusters that have a lower energy than the lowest-energy clusters from the literature.

For a given cluster size and element, we compared the lowest-energy (within 1 meV/atom) cluster from the literature against the lowest-energy cluster that was new (i.e., not from the literature) to assess which had lower energy. We subtracted the energy of the new cluster from the energy of the literature cluster. In this way, we found that the Quantum Cluster Database contains 501 lowest-energy clusters that have an energy that is lower than the energy of a lowest-energy cluster from the literature (Figure 49). There are 1540 lowest-energy clusters in the database from the literature.

The Quantum Cluster Database includes 1071 new structure types or templates (i.e., relative arrangement of atoms). We determined this quantity by considering the low-energy clusters within 1 meV of the lowest-energy cluster for a given element and size. In comparison, there are 683 templates of low-energy clusters from the literature in the Quantum Cluster Database.

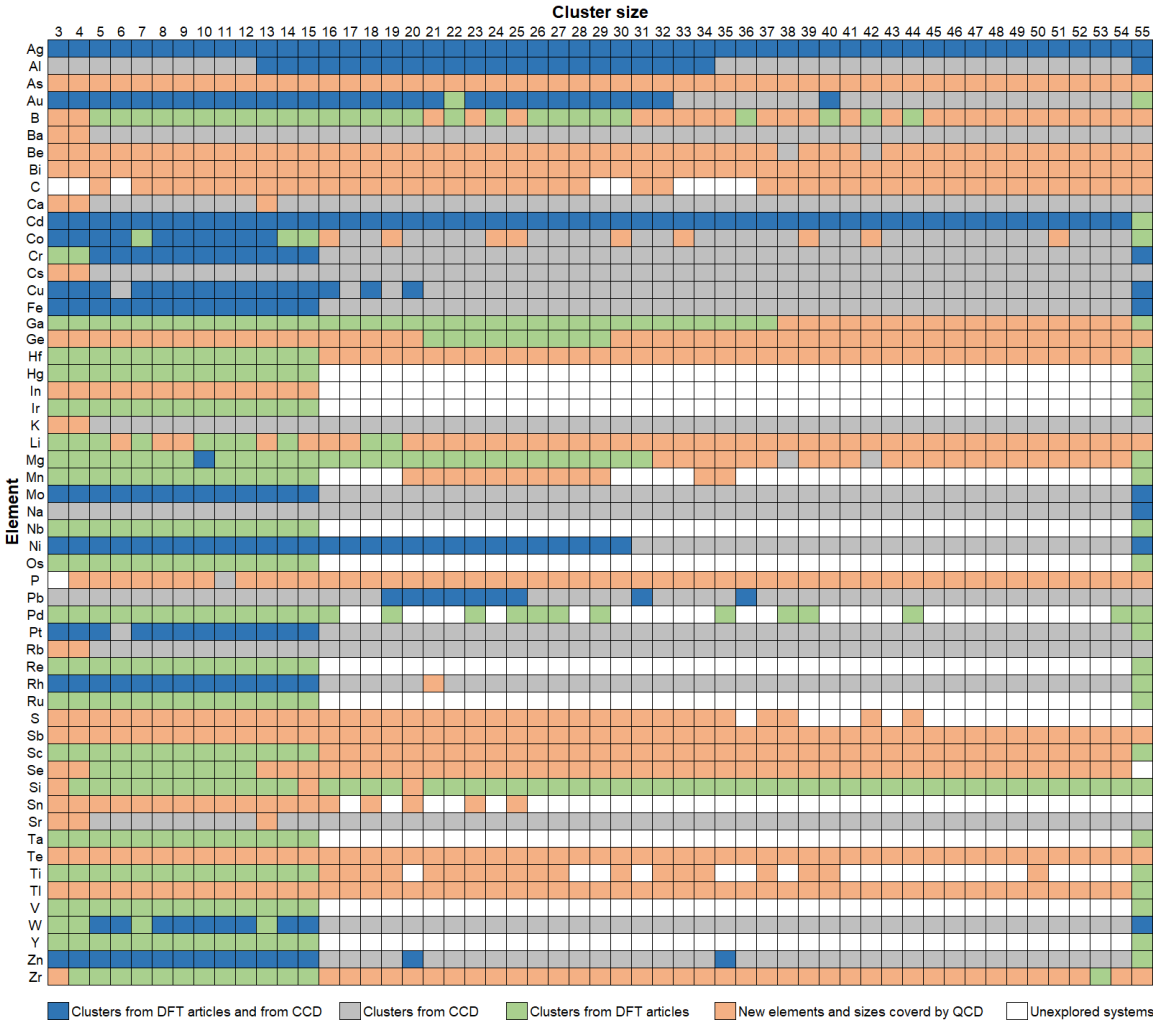


Figure 50. The Quantum Cluster Database covers 849 regions that were previously unexplored (shown in orange). A summary of previous studies of elemental clusters in terms of exploring their atomic structures. We have considered literature that used DFT to find atomic structures as well as systems covered in the Cambridge Cluster Database that used empirical potentials. We have considered 55 different elements across the periodic table with size regimes from 3-55 atoms in the cluster. Note: even when the cluster of a particular element and size has been explored in the literature, we may have discovered new clusters for that element and size.

Before our work, here were 1540 cluster elements and sizes covered by the literature out of 2915, which corresponds to 53%. With the Quantum Cluster Database, the percentage covered increased to 82%.

4.5 Usage notes

The Quantum Cluster Database is extensive in two important ways: (1) The results of DFT calculations performed on both metal and non-metal systems are reported. Previous

compilations report results obtained through lower levels of theory or only investigate specific regions of the periodic table, for example transition metals. (2) All configurational isomers for each element available in the database are included. This is in contrast with other compilations where only the most stable ground state cluster structures were studied. This should aid in both data mining and the identification of metastable, but experimentally realizable, structures. This database can be systematically improved through continuous updates, which may include the addition of more physical properties, charged systems, larger clusters, multi-element clusters, ligated systems, and user-submitted systems.

When clicking on an element, the website interface enables the visualization of the correlations against other elements (Figure 51 a), the visualization of the relative energies for every size, and a view of the cluster with a light purple background if the cluster is equivalent to a literature cluster (Figure 51 b). Then, when clicking on a particular cluster, the properties Table 24 are displayed (except for the raw DFT energy and the structure in POSCAR format) together with an interactive view of the cluster and options to download the XYZ structure file and references in .bib format.

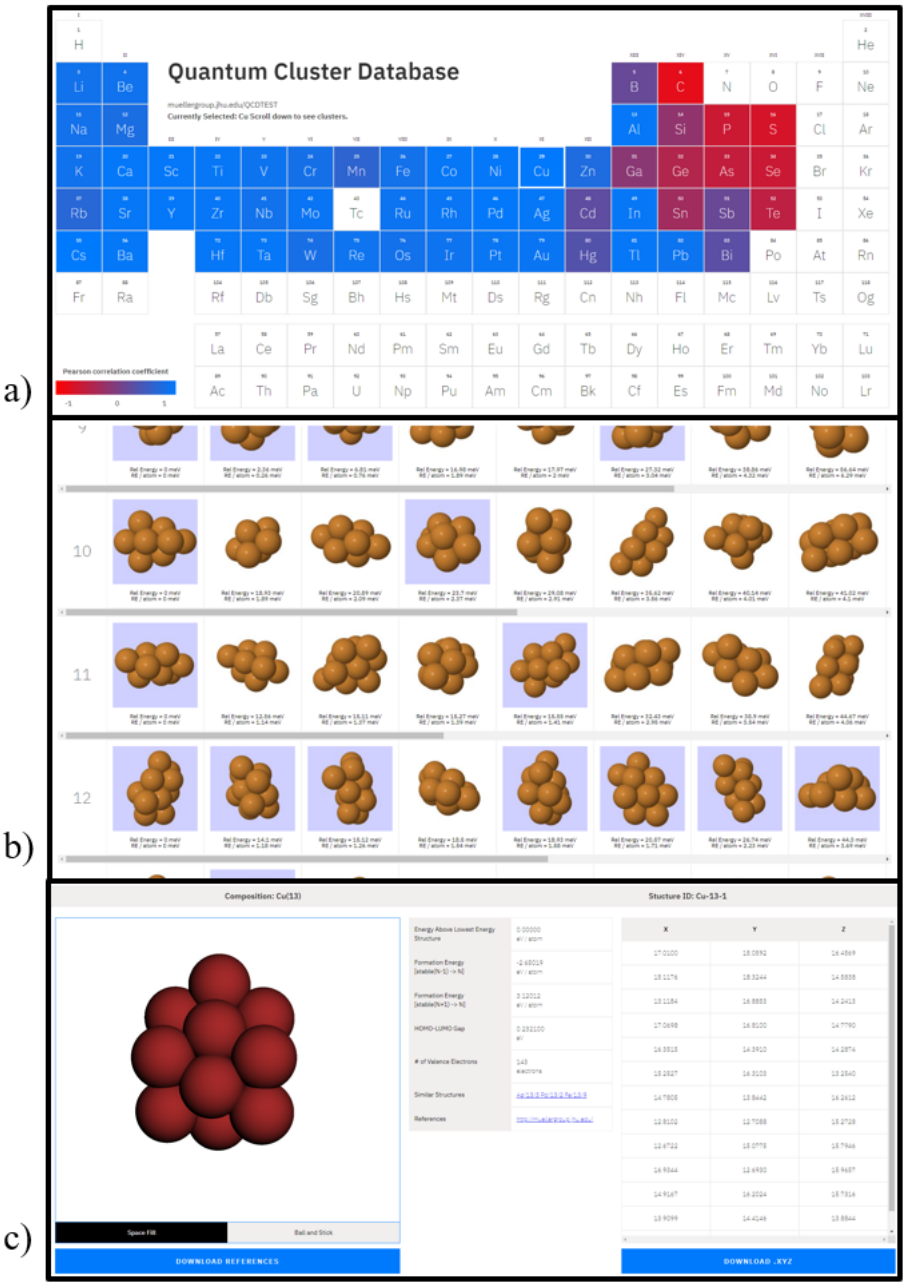


Figure 51. Interface of the Quantum Cluster Database

4.6 Code availability

The automation of the high-throughput calculations was performed using a genetic algorithm we developed, which is available via GitLab: https://gitlab.com/muellergroup/cluster-ga. VASP.

4.7 Acknowledgements

This work is supported by a grant from the Office of Naval Research, N00014-15-1-2681. Calculations were performed on a computer allocation grant provided by the Department of Defense HPC Modernization Program, STVONRDC40463482.

4.8 Author contributions

Building the Quantum Cluster Database was team effort of the Mueller Research Group. My collaborative contributions were in the areas where my name is shown. Specifically, my main contributions, where I led or co-led the work, where in the areas where my name has an asterisk next to it.

- Developed the genetic algorithm code: Phil Wang and Peter Lile
- Discovered clusters with the genetic algorithm: Sukriti Manna and Peter Lile
- Extracted clusters from the literature: Alberto Hernandez*, Sukriti Manna, and Peter Lile
- Performed DFT calculations and analyses of box size, magnetic initializations, template clusters for correlations, and correlations calculations: Sukriti Manna and Tim Mueller
- Verified that template clusters for correlations were distinct: Alberto Hernandez*

- Ran DFT to identify the equilibrium bond distance for every element: Alberto Hernandez*
- Developed and ran scripts to compute properties and metadata: Alberto Hernandez* and Sukriti Manna
- Developed and ran scripts to check DFT calculations: Alberto Hernandez* and Sukriti Manna
- Developed and ran scripts to check clusters (box size, contiguous, non-periodic): Alberto Hernandez*
- Developed and ran scripts to fix structures for re-running DFT: Alberto Hernandez*
- Developed and ran scripts to filter unique clusters (including similarity calculator) and tagged literature clusters: Alberto Hernandez*, and Tim Mueller
- Developed the website interface core code: outside team.
- Edited and installed website interface: Alberto Hernandez* and outside team
- Generated data product files (.csv and .json): Alberto Hernandez*
- Processed data, generated graphs, and analyzed the graphs: Alberto Hernandez*, Sukriti Manna, and Tim Mueller
- Conceived and managed the project: Tim Mueller

5 Conclusions and Outlook

5.1 Interatomic potential models by symbolic regression

The development of interatomic potential models has traditionally consisted of deriving simple parameterized functions from fundamental physical relationships; these models are called empirical potentials. Another more recent approach consists of developing interatomic potential models using supervised machine learning. Interatomic potential models developed through the traditional approach have several advantages. For example, they have relatively good transferability because of their physical foundations, they are simple, can require a small number of training examples, and usually are orders of magnitude faster than machine learning potential models. However, empirical potentials typically cannot achieve very high levels of accuracy due to the small hypothesis space from which they are built. Machine learning potential can be developed for a wide variety of chemical systems and achieve a high interpolative accuracy due to the large hypothesis space from which they are built, but they are orders of magnitude slower than EAM-type potentials. There is a need for interatomic potentials that are simple (and interpretable), computationally fast (for simulations at large time and length scales), that have a high accuracy on the system of interest, and that transfer well.

The symbolic regression approach presented in this work succeeded at finding simple (and interpretable), fast, accurate and transferrable (to test data and to atomic environments unlike the ones used for training) interatomic potential models for Cu, Ag, Ni and Pd from *ab initio* data, and the models developed for Cu transfer well to Ag, Ni and Pd (which are close to Cu on the periodic table). A key component of our technique is the creation of a physics-informed hypothesis space that encodes physical information into the machine

learning models. Another important component was the direct minimization of a multi-objective loss function in the space of mathematical expressions and parameters (i.e., multi-objective symbolic regression) to search simple, accurate and fast interatomic potential models. The interatomic potential models that we developed are 2-3 orders of magnitude faster than other machine learning interatomic potentials, they are on average one order of magnitude simpler than other EAM-type models, they achieve a greater accuracy on the training data than benchmark empirical potentials, and their transferability at least as good as other EAM-type models. The simplicity of the models that we developed allow the interpretation of their functional forms to possibly gain insights about the physical interactions in the atomic system that correspond to the shape of the potential energy surface. For example, our models tend to follow the embedded atom method framework, and one of them has a functional form for which the embedding function is a weighted average.

These positive results suggest that this symbolic regression approach could become a powerful tool for developing new interatomic potential models for simulations at large time and length scales, but our approach has several opportunities for improvement. Some important extensions to our algorithm are:

- Account for directional bonding in the description of the local atomic environment; essential for systems where the covalent component of bonding is important.
- Account for long-range interactions. The graph representation used by our approach can consider long-range interactions through nested summations over neighbors. The ionic component of bonding might also be included using this extension.

- Enable the development of interatomic potential modes for multi-component systems.
- Implement the optimization of the cutoff distance (i.e., inner and outer cutoff radii)
- Enable the algorithm to search or choose among various smoothing functions.

It would also be good to assess the benefit of expanding the hypothesis space with functions like the natural exponential, the natural logarithm, and trigonometric functions. On the supervised machine learning side, our current implementation of symbolic regression is through genetic programming, which uses a genetic algorithm as the learning engine, but there are other learning algorithms that can be explored to increase the efficiency or effectiveness of the search, especially when increasing the size of the hypothesis space; for example, through the addition of descriptors of the atomic environment or when considering multi-component systems.

5.2 Developing a database of atomically precise nanoclusters

Current datasets of atomically precise nanoclusters are either unavailable to the public, limited in scope, or primarily use lower levels of theory than *ab initio* methods. There is a need and interest in predicting the structures and properties of atomically precise nanoclusters at *ab initio* levels of theory (like DFT). Our approach to identify low-energy atomically precise nanoclusters with a genetic algorithm enabled the development of most extensive set of these kind of structures at the DFT level of theory, with more than 50,000 structures and their properties. Some of the future work for extending the database may focus on using machine learning interatomic potentials to search low-energy clusters, and

then computing their properties with DFT, or on considering other types of clusters (e.g., charged, supported, or ligated clusters).

6 References

1. Correa-Baena, J.-P.; Hippalgaonkar, K.; van Duren, J.; Jaffer, S.; Chandrasekhar, V. R.; Stevanovic, V.; Wadia, C.; Guha, S.; Buonassisi, T., Accelerating Materials Development via Automation, Machine Learning, and High-Performance Computing. *Joule* **2018,** *2* (8), 1410-1420.

2. Eagar, T. W., Bringing new materials to market. *Technology Review* **1995,** *98*.

3. de Pablo, J. J.; Jackson, N. E.; Webb, M. A.; Chen, L.-Q.; Moore, J. E.; Morgan, D.; Jacobs, R.; Pollock, T.; Schlom, D. G.; Toberer, E. S.; Analytis, J.; Dabo, I.; DeLongchamp, D. M.; Fiete, G. A.; Grason, G. M.; Hautier, G.; Mo, Y.; Rajan, K.; Reed, E. J.; Rodriguez, E.; Stevanovic, V.; Suntivich, J.; Thornton, K.; Zhao, J.-C., New frontiers for the materials genome initiative. *npj Computational Materials* **2019,** *5* (1), 41.

4. Mueller, T.; Kusne, A. G.; Ramprasad, R., Machine Learning in Materials Science: Recent Progress and Emerging Applications. In *Reviews in Computational Chemistry*, Parrill, A. L.; Lipkowitz, K. B., Eds. John Wiley & Sons, Inc.: 2016; Vol. 29.

5. Ramprasad, R.; Batra, R.; Pilania, G.; Mannodi-Kanakkithodi, A.; Kim, C., Machine learning in materials informatics: recent applications and prospects. *npj Computational Materials* **2017,** *3* (1), 54.

6. Rajan, K., Materials Informatics: The Materials "Gene" and Big Data. *Annual Review of Materials Research* **2015,** *45* (1), 153-169.

7. Schmidt, J.; Marques, M. R. G.; Botti, S.; Marques, M. A. L., Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials* **2019,** *5* (1), 83.

8. Plimpton, S. J.; Thompson, A. P., Computational aspects of many-body potentials. *MRS Bulletin* **2012,** *37* (5), 513-521.

9. Tadmor, E. B.; Miller, R. E., *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*. Cambridge University Press: 2011.

10. Chen, Y.; Manna, S.; Ciobanu, C. V.; Reimanis, I. E., Thermal regimes of Li-ion conductivity in β-eucryptite. *Journal of the American Ceramic Society* **2018,** *101* (1), 347-355.

11. Wang, C.; Aoyagi, K.; Wisesa, P.; Mueller, T., Lithium Ion Conduction in Cathode Coating Materials from On-the-Fly Machine Learning. *Chemistry of Materials* **2020,** *32* (9), 3741-3752.

12. Chen, Y.; Manna, S.; Narayanan, B.; Wang, Z.; Reimanis, I. E.; Ciobanu, C. V., Pressure-induced phase transformation in β-eucryptite: An X-ray diffraction and density functional theory study. *Scripta Materialia* **2016,** *122*, 64-67.

13. Li, C.; Nilson, T.; Cao, L.; Mueller, T., Predicting activation energies for vacancy-mediated diffusion in alloys using a transition-state cluster expansion. *Physical Review Materials* **2021,** *5* (1), 013803.

14. Kohn, W.; Sham, L. J., Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **1965,** *140*, A1133-A1138.

15. Hohenberg, P.; Kohn, W., Inhomogeneous Electron Gas. *Phys. Rev.* **1964,** *136*, 864.

16. Goedecker, S., Linear scaling electronic structure methods. *Reviews of Modern Physics* **1999,** *71* (4), 1085-1123.

17. Mohr, S.; Eixarch, M.; Amsler, M.; Mantsinen, M. J.; Genovese, L., Linear scaling DFT calculations for large tungsten systems using an optimized local basis. *Nuclear Materials and Energy* **2018,** *15*, 64-70.

18. Shao, Y.; Molnar, L. F.; Jung, Y.; Kussmann, J.; Ochsenfeld, C.; Brown, S. T.; Gilbert, A. T. B.; Slipchenko, L. V.; Levchenko, S. V.; O'Neill, D. P.; DiStasio, R. A.; Lochan, R. C.; Wang, T.; Beran, G. J. O.; Besley, N. A.; Herbert, J. M.; Lin, C. Y.; Van Voorhis, T.; Chien, S. H.; Sodt, A.; Steele, R. P.; Rassolov, V. A.; Maslen, P. E.; Korambath, P. P.; Adamson, R. D.; Austin, B.; Baker, J.; Byrd, E. F. C.; Dachsel, H.; Doerksen, R. J.; Dreuw, A.; Dunietz, B. D.; Dutoi, A. D.; Furlani, T. R.; Gwaltney, S. R.; Heyden, A.; Hirata, S.; Hsu, C. P.; Kedziora, G.; Khalliulin, R. Z.; Klunzinger, P.; Lee, A. M.; Lee, M. S.; Liang, W.; Lotan, I.; Nair, N.; Peters, B.; Proynov, E. I.; Pieniazek, P. A.; Rhee, Y. M.; Ritchie, J.; Rosta, E.; Sherrill, C. D.; Simmonett, A. C.; Subotnik, J. E.; Woodcock, H. L.; Zhang, W.; Bell, A. T.; Chakraborty, A. K.; Chipman, D. M.; Keil, F. J.; Warshel, A.; Hehre, W. J.; Schaefer, H. F.; Kong, J.; Krylov, A. I.; Gill, P. M. W.; Head-Gordon, M., Advances in methods and algorithms in a modern quantum chemistry program package. *Physical Chemistry Chemical Physics* **2006,** *8* (27), 3172-3191.

19. Needs, R. J.; Towler, M. D.; Drummond, N. D.; Rios, P. L., Continuum variational and diffusion quantum Monte Carlo calculations. *J. Phys.-Condes. Matter* **2010,** *22* (2).

20. Saritas, K.; Mueller, T.; Wagner, L.; Grossman, J. C., Investigation of a Quantum Monte Carlo Protocol To Achieve High Accuracy and High-Throughput Materials Formation Energies. *Journal of Chemical Theory and Computation* **2017,** *13* (5), 1943-1951.

21. Coulomb, C. A., *Mémoires sur l'électricité et la magnétisme*. Chez Bachelier, libraire: 1789.

22. Lennard-Jones, J. E., *Proc. Phys. Soc.* **1931,** *43*, 461.

23. Buckingham, R. A.; Lennard-Jones, J. E., The classical equation of state of gaseous helium, neon and argon. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **1938,** *168* (933), 264-283.

24. Stillinger, F. H.; Weber, T. A., Computer simulation of local order in condensed phases of silicon. *Physical Review B* **1985,** *31* (8), 5262-5271.

25. Daw, M. S.; Baskes, M. I., Embedded-atom method: {Derivation} and application to impurities, surfaces, and other defects in metals. *Physical Review B* **1984,** *29*, 6443.

26. Brenner, D. W.; Shenderova, O. A.; Areshkin, D. A., Quantum-Based Analytic Interatomic Forces and Materials Simulation. In *Reviews in Computational Chemistry*, 1998; pp 207-239.

27. Brenner, D. W., Relationship between the embedded-atom method and Tersoff potentials. *Physical Review Letters* **1989,** *63* (9), 1022-1022.

28. Baskes, M. I., Application of the Embedded-Atom Method to Covalent Materials: A Semiempirical Potential for Silicon. *Physical Review Letters* **1987,** *59* (23), 2666-2669.

29. Lee, B.-J.; Shim, J.-H.; Baskes, M. I., Semiempirical atomic potentials for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, Al, and Pb based on first and second nearest-neighbor modified embedded atom method. *Physical Review B* **2003,** *68* (14), 144112.

30. Mishin, Y.; Mehl, M. J.; Papaconstantopoulos, D. A., Phase stability in the Fe–Ni system: Investigation by first-principles calculations and atomistic simulations. *Acta Materialia* **2005,** *53* (15), 4029-4041.

31. Tersoff, J., New empirical approach for the structure and energy of covalent systems. *Physical Review B* **1988,** *37* (12), 6991-7000.

32. Liang, T.; Shan, T.-R.; Cheng, Y.-T.; Devine, B. D.; Noordhoek, M.; Li, Y.; Lu, Z.; Phillpot, S. R.; Sinnott, S. B., Classical atomistic simulations of surfaces and heterogeneous interfaces with the charge-optimized many body (COMB) potentials. *Materials Science and Engineering: R: Reports* **2013,** *74* (9), 255-279.

33. Shan, T.-R.; Devine, B. D.; Hawkins, J. M.; Asthagiri, A.; Phillpot, S. R.; Sinnott, S. B., Second-generation charge-optimized many-body potential for $\text{Si}/{\text{SiO}}_{2}$ and amorphous silica. *Physical Review B* **2010,** *82* (23), 235302.

34. Yu, J.; Sinnott, S. B.; Phillpot, S. R., Charge optimized many-body potential for the $\mathrm{Si}/{\mathrm{SiO}}_{2}$ system. *Physical Review B* **2007,** *75* (8), 085311.

35. Senftle, T. P.; Hong, S.; Islam, M. M.; Kylasa, S. B.; Zheng, Y.; Shin, Y. K.; Junkermeier, C.; Engel-Herbert, R.; Janik, M. J.; Aktulga, H. M.; Verstraelen, T.; Grama, A.; van Duin, A. C. T., The ReaxFF reactive force-field: development, applications and future directions. *Npj Computational Materials* **2016,** *2*, 15011.

36. van Duin, A. C. T.; Dasgupta, S.; Lorant, F.; Goddard, W. A., ReaxFF: A reactive force field for hydrocarbons. *J. Phys. Chem. A* **2001,** *105* (41), 9396-9409.

37. Becker, C. A.; Tavazza, F.; Trautt, Z. T.; Buarque de Macedo, R. A., Considerations for choosing and using force fields and interatomic potentials in materials science and engineering. *Current Opinion in Solid State and Materials Science* **2013,** *17* (6), 277-283.

38. Hynninen, T.; Musso, T.; Foster, A. S., Limitations of reactive atomistic potentials in describing defect structures in oxides. *Modelling and Simulation in Materials Science and Engineering* **2016,** *24* (3), 035022.

39. Iype, E.; Hütter, M.; Jansen, A. P. J.; Nedea, S. V.; Rindt, C. C. M., Parameterization of a reactive force field using a Monte Carlo algorithm. *Journal of Computational Chemistry* **2013,** *34* (13), 1143-1154.

40. Trnka, T.; Tvaroška, I.; Koča, J., Automated Training of ReaxFF Reactive Force Fields for Energetics of Enzymatic Reactions. *Journal of Chemical Theory and Computation* **2018,** *14* (1), 291-302.

41. Chan, H.; Narayanan, B.; Cherukara, M. J.; Sen, F. G.; Sasikumar, K.; Gray, S. K.; Chan, M. K. Y.; Sankaranarayanan, S. K. R. S., Machine Learning Classical Interatomic

Potentials for Molecular Dynamics from First-Principles Training Data. *The Journal of Physical Chemistry C* **2019,** *123* (12), 6941-6957.

42. Li, Y.; Li, H.; Pickard, F. C.; Narayanan, B.; Sen, F. G.; Chan, M. K. Y.; Sankaranarayanan, S. K. R. S.; Brooks, B. R.; Roux, B., Machine Learning Force Field Parameters from Ab Initio Data. *Journal of Chemical Theory and Computation* **2017,** *13* (9), 4492-4503.

43. Pahari, P.; Chaturvedi, S., Determination of best-fit potential parameters for a reactive force field using a genetic algorithm. *Journal of Molecular Modeling* **2012,** *18* (3), 1049-1061.

44. Larsson, H. R.; van Duin, A. C. T.; Hartke, B., Global optimization of parameters in the reactive force field ReaxFF for SiOH. *Journal of Computational Chemistry* **2013,** *34* (25), 2178-2189.

45. Sen, F. G.; Kinaci, A.; Narayanan, B.; Gray, S. K.; Davis, M. J.; Sankaranarayanan, S. K. R. S.; Chan, M. K. Y., Towards accurate prediction of catalytic activity in IrO2 nanoclusters via first principles-based variable charge force field. *Journal of Materials Chemistry A* **2015,** *3* (37), 18970-18982.

46. Cherukara, M. J.; Narayanan, B.; Kinaci, A.; Sasikumar, K.; Gray, S. K.; Chan, M. K. Y.; Sankaranarayanan, S. K. R. S., Ab Initio-Based Bond Order Potential to Investigate Low Thermal Conductivity of Stanene Nanostructures. *The Journal of Physical Chemistry Letters* **2016,** *7* (19), 3752-3759.

47. Ercolessi, F.; Adams, J. B., Interatomic Potentials from First-Principles Calculations: The Force-Matching Method. *EPL (Europhysics Letters)* **1994,** *26* (8), 583.

48. Widrow, B.; Lehr, M. A., 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE* **1990,** *78* (9), 1415-1442.

49. Podryabinkin, E. V.; Shapeev, A. V., Active learning of linearly parametrized interatomic potentials. *Computational Materials Science* **2017,** *140*, 171-180.

50. Yunxing Zuo, C. C., Xiangguo Li, Zhi Deng, Yiming Chen, Jörg Behler, Gábor Csányi, Alexander V. Shapeev, Aidan P. Thompson, Mitchell A. Wood, Shyue Ping Ong, A Performance and Cost Assessment of Machine Learning Interatomic Potentials. *arXiv:1906.08888v3 [physics.comp-ph]* **2019**.

51. Yao, K.; Herr, J. E.; Toth, David W.; McKintyre, R.; Parkhill, J., The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics. *Chemical Science* **2018,** *9* (8), 2261-2269.

52. Dragoni, D.; Daff, T. D.; Csányi, G.; Marzari, N., Achieving DFT accuracy with a machine-learning interatomic potential: Thermomechanics and defects in bcc ferromagnetic iron. *Physical Review Materials* **2018,** *2* (1), 013808.

53. Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R., Machine learning of accurate energy-conserving molecular force fields. *Science Advances* **2017,** *3* (5), e1603015.

54. Huan, T. D.; Batra, R.; Chapman, J.; Krishnan, S.; Chen, L.; Ramprasad, R., A universal strategy for the creation of machine learning-based atomistic force fields. *npj Computational Materials* **2017,** *3* (1), 37.

55. Tang, Y.-H.; Zhang, D.; Karniadakis, G. E., An atomistic fingerprint algorithm for learning ab initio molecular force fields. *The Journal of Chemical Physics* **2018,** *148* (3), 034101.

56. Behler, J.; Parrinello, M., Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters* **2007,** *98* (14), 146401.

57. Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G., Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **2010,** *104*, 136403.

58. Thompson, A. P.; Swiler, L. P.; Trott, C. R.; Foiles, S. M.; Tucker, G. J., Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics* **2015,** *285*, 316-330.

59. Wood, M. A.; Thompson, A. P.; W., B. D., Extending the accuracy of the SNAP interatomic potential form. *The Journal of Chemical Physics* **2018,** *148* (24), 241721.

60. Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A., Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters* **2012,** *108* (5), 058301.

61. Batra, R.; Tran, H. D.; Kim, C.; Chapman, J.; Chen, L.; Chandrasekaran, A.; Ramprasad, R., General Atomic Neighborhood Fingerprint for Machine Learning-Based Methods. *The Journal of Physical Chemistry C* **2019,** *123* (25), 15859-15866.

62. Drautz, R., Atomic cluster expansion for accurate and transferable interatomic potentials. *Physical Review B* **2019,** *99* (1), 014104.

63. Seko, A.; Togo, A.; Tanaka, I., Group-theoretical high-order rotational invariants for structural representations: Application to linearized machine learning interatomic potential. *Physical Review B* **2019,** *99* (21), 214108.

64. Artrith, N.; Urban, A.; Ceder, G., Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Physical Review B* **2017,** *96* (1), 014112.

65. Seko, A.; Takahashi, A.; Tanaka, I., First-principles interatomic potentials for ten elemental metals via compressed sensing. *Physical Review B* **2015,** *92* (5), 054113.

66. Oganov, A. R.; Valle, M., How to quantify energy landscapes of solids. *The Journal of Chemical Physics* **2009,** *130* (10), 104504.

67. Jacobsen, T. L.; Jørgensen, M. S.; Hammer, B., On-the-Fly Machine Learning of Atomic Potential in Density Functional Theory Structure Optimization. *Physical Review Letters* **2018,** *120* (2), 026102.

68. Bartók, A. P.; Kondor, R.; Csányi, G., On representing chemical environments. *Physical Review B* **2013,** *87* (18), 184115.

69. Behler, J., Perspective: Machine learning potentials for atomistic simulations. *The Journal of Chemical Physics* **2016,** *145* (17), 170901.

70. Behler, J., First Principles Neural Network Potentials for Reactive Simulations of Large Molecular and Condensed Systems. *Angewandte Chemie International Edition* **2017,** *56* (42), 12828-12840.

71. Shapeev, A., Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials. *Multiscale Modeling & Simulation* **2016,** *14* (3), 1153-1173.

72. Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints *arXiv e-prints* [Online], 2015. (accessed September 01, 2015).

73. Bjørn Jørgensen, P.; Wedel Jacobsen, K.; Schmidt, M. N. Neural Message Passing with Edge Updates for Predicting Properties of Molecules and Materials *arXiv e-prints* [Online], 2018. (accessed June 01, 2018).

74. Xie, T.; Grossman, J. C., Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Physical Review Letters* **2018,** *120* (14), 145301.

75. Schütt, K. T.; Sauceda, H. E.; Kindermans, P. J.; Tkatchenko, A.; Müller, K. R., SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics* **2018,** *148* (24), 241722.

76. Justin Gilmer, S. S. S., Patrick F. Riley, Oriol Vinyals, George E. Dahl, Neural Message Passing for Quantum Chemistry. *arXiv:1704.01212 [cs.LG]* **2017**.

77. Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P., Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design* **2016,** *30* (8), 595-608.

78. Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A., Quantum-chemical insights from deep tensor neural networks. *Nature Communications* **2017,** *8*, 13890.

79. Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P., Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chemistry of Materials* **2019,** *31* (9), 3564-3572.

80. Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P., Convolutional networks on graphs for learning molecular fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, MIT Press: Montreal, Canada, 2015; pp 2224–2232.

81. Cheon, G.; Yang, L.; McCloskey, K.; Reed, E. J.; Cubuk, E. D., Crystal Structure Search with Random Relaxations Using Graph Networks. *arXiv:2012.02920 [cond-mat.mtrl-sci]* **2020**.

82. Mueller, T.; Hernandez, A.; Wang, C., Machine learning for interatomic potential models. *The Journal of Chemical Physics* **2020,** *152* (5), 050902.

83. Schmidt, M.; Lipson, H., Distilling Free-Form Natural Laws from Experimental Data. *Science* **2009,** *324* (5923), 81-85.

84. Udrescu, S.-M.; Tegmark, M., AI Feynman: A physics-inspired method for symbolic regression. *Science Advances* **2020,** *6* (16), eaay2631.

85. Li, L.; Fan, M.; Singh, R.; Riley, P., Neural-Guided Symbolic Regression with Asymptotic Constraints. *arXiv:1901.07714 [cs.LG]* **2019**.

86. Bongard, J.; Lipson, H., Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **2007,** *104* (24), 9943.

87. Abdel Kenoufi, K. T. K., Symbolic Regression of Inter-Atomic Potentials via Genetic Programming. *Biological and Chemical Research* **2015,** *2* (1), 1-15.

88. Smits, G. F.; Kotanchek, M., Pareto-front exploitation in symbolic regression. In *Genetic Programming Theory and Practice II*, Springer: 2005; pp 283-299.

89. Hernandez, A.; Balasubramanian, A.; Yuan, F.; Mason, S. A. M.; Mueller, T., Fast, accurate, and transferable many-body interatomic potentials by symbolic regression. *npj Computational Materials* **2019,** *5* (1), 112.

90. Yuan, F.; Mueller, T., Identifying models of dielectric breakdown strength from high-throughput data via genetic programming. *Scientific Reports* **2017,** *7* (1), 17594.

91. Slepoy, A.; Peters, M. D.; Thompson, A. P., Searching for globally optimal functional forms for interatomic potentials using genetic programming with parallel tempering. *Journal of Computational Chemistry* **2007,** *28* (15), 2465-2471.

92. Makarov, D. E.; Metiu, H., Fitting potential-energy surfaces: A search in the function space by directed genetic programming. *The Journal of Chemical Physics* **1998,** *108* (2), 590-598.

93. Neely, C.; Weller, P.; Dittmar, R., Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *J. Financ. Quant. Anal.* **1997,** *32* (4), 405-426.

94. Brown, W. M.; Thompson, A. P.; Schultz, P. A., Efficient hybrid evolutionary optimization of interatomic potential models. *The Journal of Chemical Physics* **2010,** *132*, 24108.

95. Karaboga, D.; Ozturk, C.; Karaboga, N.; Gorkemli, B., Artificial bee colony programming for symbolic regression. *Information Sciences* **2012,** *209*, 1-15.

96. Icke, I.; Bongard, J. C. In *Improving genetic programming based symbolic regression using deterministic machine learning*, 2013 IEEE Congress on Evolutionary Computation, 20-23 June 2013; 2013; pp 1763-1770.

97. Poli, R.; Langdon, W. B.; McPhee, N. F., *A Field Guide to Genetic Programming*. 2008.

98. Goldberg, D. E.; Holland, J. H., Genetic Algorithms and Machine Learning. *Machine Learning* **1988,** *3* (2), 95-99.

99. Holland, J. H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. 1 ed.; Ann Arbor, MI: University of Michigan Press: 1975.

100. Koza, J. R., *Genetic programming – on the programming of computers by means of natural selection*. MIT Press: 1992.

101. Ercolessi, F.; Adams, J. B., Interatomic Potentials from First-Principles Calculations: The Force-Matching Method. *Europhysics Letters (EPL)* **1994,** *26* (8), 583-588.

102. Balabin, R. M.; Lomakina, E. I., Support vector machine regression (LS-SVM)—an alternative to artificial neural networks (ANNs) for the analysis of quantum chemistry data? *Physical Chemistry Chemical Physics* **2011,** *13* (24), 11710-11718.

103. Mueller, T.; Ceder, G., Bayesian approach to cluster expansions. *Physical Review B* **2009,** *80* (2), 024103.

104. Li, Z.; Kermode, J. R.; De Vita, A., Molecular Dynamics with On-the-Fly Machine Learning of Quantum-Mechanical Forces. *Physical Review Letters* **2015,** *114* (9), 096405.

105. Botu, V.; Ramprasad, R., Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International Journal of Quantum Chemistry* **2015,** *115* (16), 1074-1083.

106. Smith, J. S.; Isayev, O.; Roitberg, A. E., ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical Science* **2017,** *8* (4), 3192-3203.

107. Cao, L.; Li, C.; Mueller, T., The Use of Cluster Expansions To Predict the Structures and Properties of Surfaces and Nanostructured Materials. *Journal of Chemical Information and Modeling* **2018,** *58* (12), 2401-2413.

108. Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W., Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. *Physical Review Letters* **2018,** *120* (14), 143001.

109. Nyshadham, C.; Rupp, M.; Bekker, B.; Shapeev, A. V.; Mueller, T.; Rosenbrock, C. W.; Csányi, G.; Wingate, D. W.; Hart, G. L. W., Machine-learned multi-system surrogate models for materials prediction. *npj Computational Materials* **2019,** *5* (1), 51.

110. Daw, M. S.; Baskes, M. I., Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Physical Review B* **1984,** *29* (12), 6443-6453.

111. Finnis, M. W.; Sinclair, J. E., A simple empirical N-body potential for transition metals. *Philosophical Magazine A* **1984,** *50* (1), 45-55.

112. F. Ercolessi, M. P., E. Tosatti, Simulation of gold in the glue model. *Philosophical Magazine A* **1988,** *58* (1), 213-226.

113. Brenner, D. W.; Shenderova, O. A.; Areshkin, D. A., Quantum-Based Analytic Interatomic Forces and Materials Simulation. *Reviews in Computational Chemistry* **1998**.

114. Sinnott, S. B.; Brenner, D. W., Three decades of many-body potentials in materials research. *MRS Bulletin* **2012,** *37* (5), 469-473.

115. Finnis, M. W., Concepts for simulating and understanding materials at the atomic scale. *MRS Bulletin* **2012,** *37* (5), 477-484.

116. Tim Mueller, A. G. K., and Rampi Ramprasad, Reviews in Computational Chemistry. In *Reviews in Computational Chemistry*, Lipkowitz, A. L. P. a. K. B., Ed. John Wiley & Sons, Inc.: 2016; Vol. 29, p 188.

117. Mueller, T.; Johlin, E.; Grossman, J. C., Origins of hole traps in hydrogenated nanocrystalline and amorphous silicon revealed through machine learning. *Physical Review B* **2014,** *89* (11), 115202.

118. Lennard-Jones, J. E., Cohesion. *Proceedings of the Physical Society* **1931,** *43* (5), 461-482.

119. Morse, P. M., Diatomic Molecules According to the Wave Mechanics. II. Vibrational Levels. *Physical Review* **1929,** *34* (1), 57-64.

120. Abell, G. C., Empirical chemical pseudopotential theory of molecular and metallic bonding. *Physical Review B* **1985,** *31* (10), 6184-6196.

121. Hawkins, D. M., The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences* **2004,** *44* (1), 1-12.

122. Ding, H.-Q.; Karasawa, N.; Goddard, W. A., Optimal spline cutoffs for Coulomb and van der Waals interactions. *Chemical Physics Letters* **1992,** *193*, 197-201.

123. Mishin, Y.; Mehl, M. J.; Papaconstantopoulos, D. A.; Voter, A. F.; Kress, J. D., Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations. *Physical Review B* **2001,** *63* (22), 224106.

124. Jeffrey Horn, N. N., and David E. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization. *Proc. First IEEE Conf. Evolutionary Computation* **1994**, 82-87.

125. Ekárt, A.; Németh, S. Z., Selection Based on the Pareto Nondomination Criterion for Controlling Code Growth in Genetic Programming. *Genetic Programming and Evolvable Machines* **2001,** *2*, 61-73.

126. Hansen, N.; Ostermeier, A., Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996; pp 312-317.

127. Polak, E.; Ribiere, G., Note surla convergence des methodes de directions conjuguees. *Imform. Rech. Oper.* **1969,** *16*, 35.

128. Kresse, G.; Furthmuller, J., Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **1996,** *54*, 11169.

129. Perdew, J. P.; Burke, K.; Ernzerhof, M., Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996,** *77*, 3865-3868.

130. Blöchl, P. E., Projector augmented-wave method. *Phys. Rev. B* **1994,** *50*, 17953-17979.

131. Wisesa, P.; McGill, K. A.; Mueller, T., Efficient generation of generalized Monkhorst-Pack grids through the use of informatics. *Phys. Rev. B* **2016,** *93*, 155109.

132. Angsten, T.; Mayeshiba, T.; Wu, H.; Morgan, D., Elemental vacancy diffusion database from high-throughput first-principles calculations for fcc and hcp structures. *New Journal of Physics* **2014,** *16* (1), 015018.

133. Tran, R.; Xu, Z.; Radhakrishnan, B.; Winston, D.; Sun, W.; Persson, K. A.; Ong, S. P., Surface energies of elemental crystals. *Scientific Data* **2016,** *3*, 160080.

134. Wulff, G., Zur Frage der Geschwindigkeit des Wachstums und der Auflösung der Krystallflagen. *Z. Kryst. Mineral.* **1901,** *34*.

135. Ong, S. P.; Richards, W. D.; Jain, A.; Hautier, G.; Kocher, M.; Cholia, S.; Gunter, D.; Chevrier, V. L.; Persson, K. A.; Ceder, G., Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science* **2013,** *68*, 314-319.

136. Onat, B.; Durukanoğlu, S., An optimized interatomic potential for Cu–Ni alloys with the embedded-atom method. *Journal of Physics: Condensed Matter* **2013,** *26* (3), 035404.

137. Plimpton, S., Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics* **1995,** *117*, 1-19.

138. Sutton, A. P.; Chen, J., Long-range Finnis–Sinclair potentials. *Philosophical Magazine Letters* **1990,** *61*, 139-146.

139. Rose, J. H.; Smith, J. R.; Guinea, F.; Ferrante, J., Universal features of the equation of state of metals. *Physical Review B* **1984,** *29* (6), 2963-2969.

140. Mendelev, M. I.; Kramer, M. J.; Becker, C. A.; Asta, M., Analysis of semi-empirical interatomic potentials appropriate for simulation of crystalline and liquid Al and Cu. *Philosophical Magazine* **2008,** *88*, 1723-1750.

141. Ackland, G. J.; Bacon, D. J.; Calder, A. F.; Harry, T., Computer simulation of point defect properties in dilute Fe—Cu alloy using a many-body interatomic potential. *Philosophical Magazine A* **1997,** *75* (3), 713-732.

142. Mendelev, M. I.; King, A. H., The interactions of self-interstitials with twin boundaries. *Philosophical Magazine* **2013,** *93* (10-12), 1268-1278.

143. Adams, J. B.; Foiles, S. M.; Wolfer, W. G., Self-diffusion and impurity diffusion of fee metals using the five-frequency model and the Embedded Atom Method. *Journal of Materials Research* **1989,** *4* (1), 102-112.

144. Foiles, S. M.; Baskes, M. I.; Daw, M. S., Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys. *Physical Review B* **1986,** *33* (12), 7983-7991.

145. McLean, A. D.; McLean, R. S., Roothaan-Hartree-Fock atomic wave functions Slater basis-set expansions for Z = 55–92. *Atomic Data and Nuclear Data Tables* **1981,** *26* (3), 197-381.

146. Clementi, E.; Roetti, C., Roothaan-Hartree-Fock atomic wavefunctions: Basis functions and their coefficients for ground and certain excited states of neutral and ionized atoms, Z≤54. *Atomic Data and Nuclear Data Tables* **1974,** *14* (3), 177-478.

147. Borovikov, V.; Mendelev, M. I.; King, A. H., Effects of stable and unstable stacking fault energy on dislocation nucleation in nano-crystalline metals. *Modelling and Simulation in Materials Science and Engineering* **2016,** *24* (8), 085017.

148. Cloutman, L. D., A Selected Library of Transport Coefficients for Combustion and Plasma Physics Applications. **2000**.

149. Hohenberg, P.; Kohn, W., Inhomogeneous Electron Gas. *Physical Review* **1964,** *136* (3B), B864-B871.

150. Chen, C.; Deng, Z.; Tran, R.; Tang, H.; Chu, I.-H.; Ong, S. P., Accurate force field for molybdenum by machine learning large materials data. *Phys. Rev. Materials* **2017,** *1*, 43603.

151. Artrith, N.; Behler, J., High-dimensional neural network potentials for metal surfaces: A prototype study for copper. *Physical Review B* **2012,** *85* (4), 045439.

152. Bartók, A. P. The Gaussian Approximation Potential: an interatomic potential derived from first principles Quantum Mechanics. Cambridge, arXiv, 2010.

153. Park, H.; Fellinger, M. R.; Lenosky, T. J.; Tipton, W. W.; Trinkle, D. R.; Rudin, S. P.; Woodward, C.; Wilkins, J. W.; Hennig, R. G., Ab initio based empirical potential used to study the mechanical properties of molybdenum. *Phys. Rev. B* **2012,** *85*, 214121.

154. Behler, J., Neural network potential-energy surfaces for atomistic simulations. In *Chemical Modelling: Applications and Theory Volume 7*, The Royal Society of Chemistry: 2010; Vol. 7, pp 1-41.

155. Deringer, V. L.; Csányi, G., Machine learning based interatomic potential for amorphous carbon. *Physical Review B* **2017,** *95* (9), 094203.

156. Behler, J., Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Physical Chemistry Chemical Physics* **2011,** *13* (40), 17930-17955.

157. Szlachta, W. J.; Bartók, A. P.; Csányi, G., Accuracy and transferability of Gaussian approximation potential models for tungsten. *Physical Review B* **2014,** *90* (10), 104108.

158. Borges, C. E.; #233; Alonso, s. L.; Jos; #233; Monta, L.; #241, Model selection in genetic programming. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, ACM: Portland, Oregon, USA, 2010; pp 985-986.

159. Burke, K., Perspective on density functional theory. *The Journal of Chemical Physics* **2012,** *136* (15), 150901.

160. Alford, R. F.; Leaver-Fay, A.; Jeliazkov, J. R.; O'Meara, M. J.; DiMaio, F. P.; Park, H.; Shapovalov, M. V.; Renfrew, P. D.; Mulligan, V. K.; Kappel, K.; Labonte, J. W.; Pacella, M. S.; Bonneau, R.; Bradley, P.; Dunbrack, R. L.; Das, R.; Baker, D.; Kuhlman, B.; Kortemme, T.; Gray, J. J., The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of Chemical Theory and Computation* **2017,** *13* (6), 3031-3048.

161. Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M., CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry* **1983,** *4* (2), 187-217.

162. Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A., Development and testing of a general amber force field. *Journal of Computational Chemistry* **2004,** *25* (9), 1157-1174.

163. Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M., UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American Chemical Society* **1992,** *114* (25), 10024-10035.

164. Lorenz, S.; Groß, A.; Scheffler, M., Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chemical Physics Letters* **2004,** *395* (4), 210-215.

165. Seko, A.; Takahashi, A.; Tanaka, I., Sparse representation for a potential energy surface. *Physical Review B* **2014,** *90* (2), 024101.

166. De Vita, A.; Car, R., A Novel Scheme for Accurate Md Simulations of Large Systems. *MRS Proceedings* **1997,** *491*, 473.

167. Pun, G. P. P.; Batra, R.; Ramprasad, R.; Mishin, Y., Physically informed artificial neural networks for atomistic modeling of materials. *Nature Communications* **2019,** *10* (1), 2339.

168. Zhang, L.; Lin, D.-Y.; Wang, H.; Car, R.; E, W., Active learning of uniformly accurate interatomic potentials for materials simulation. *Physical Review Materials* **2019,** *3* (2), 023804.

169. Auger, A.; Hansen, N. In *A restart CMA evolution strategy with increasing population size*, 2005 IEEE Congress on Evolutionary Computation, 2-5 Sept. 2005; 2005; pp 1769-1776 Vol. 2.

170. Mattsson, A. E.; Wixom, R. R.; Armiento, R., Electronic surface error in the Si interstitial formation energy. *Physical Review B* **2008,** *77* (15), 155211.

171. Nazarov, R.; Hickel, T.; Neugebauer, J., Vacancy formation energies in fcc metals: Influence of exchange-correlation functionals and correction schemes. *Physical Review B* **2012,** *85* (14), 144118.

172. O'Brien, C. J.; Barr, C. M.; Price, P. M.; Hattar, K.; Foiles, S. M., Grain boundary phase transformations in PtAu and relevance to thermal stabilization of bulk nanocrystalline metals. *Journal of Materials Science* **2018,** *53* (4), 2911-2927.

173. Balluffi, R. W., Vacancy defect mobilities and binding energies obtained from annealing studies. *Journal of Nuclear Materials* **1978,** *69-70*, 240-263.

174. Siegel, R. W., Vacancy concentrations in metals. *Journal of Nuclear Materials* **1978,** *69-70*, 117-146.

175. Ackland, G. J.; Tichy, G.; Vitek, V.; Finnis, M. W., Simple N-body potentials for the noble metals and nickel. *Philosophical Magazine A* **1987,** *56* (6), 735-756.

176. Sheng, H. W.; Kramer, M. J.; Cadien, A.; Fujita, T.; Chen, M. W., Highly optimized embedded-atom-method potentials for fourteen fcc metals. *Physical Review B* **2011,** *83* (13), 134118.

177. Hehenkamp, T.; Berger, W.; Kluin, J. E.; Lüdecke, C.; Wolff, J., Equilibrium vacancy concentrations in copper investigated with the absolute technique. *Physical Review B* **1992,** *45* (5), 1998-2003.

178. McGervey, J. D.; Triftshäuser, W., Vacancy-formation energies in copper and silver from positron annihilation. *Physics Letters A* **1973,** *44* (1), 53-54.

179. Fukushima, H.; Doyama, M., The formation energies of a vacancy in pure Cu, Cu-Si, Cu-Ga and Cu- gamma Mn solid solutions by positron annihilation. *Journal of Physics F: Metal Physics* **1976** *6*(5).

180. Triftshäuser, W.; McGervey, J. D., Monovacancy formation energy in copper, silver, and gold by positron annihilation. *Applied physics* **1975,** *6* (2), 177-180.

181. Wollenberger, H., *Physical Metallurgy*. North-Holland: Amsterdam, 1983.

182. Kimura, Y.; Qi, Y.; Cagin, T.; Goddard, W. A., The Quantum Sutton-Chen Many-body Potential for Properties of fcc Metals. In *MRS Symposium*, 1999.

183. Wycisk, W.; Feller-Kniepmeier, M., Quenching experiments in high purity Ni. *Journal of Nuclear Materials* **1978,** *69-70*, 616-619.

184. Smedskjaer, L. C.; Fluss, M. J.; Legnini, D. G.; Chason, M. K.; Siegel, R. W., The vacancy formation enthalpy in Ni determined by positron annihilation. *Journal of Physics F: Metal Physics* **1981,** *11*.

185. Hultgren, R.; Desai, P. D.; Hawkins, D. M.; Gleiser, M.; Kelley, K., *Selected Values of the Thermodynamic Properties of Binary Alloys*. American Society for Metals: 1973.

186. Schultz, H.; Ehrhart, P., *Atomic Defects in Metals,*. Springer Materials: Berlin, 1991; Vol. 25.

187. Emrick, R. M., The formation volume and energy of single vacancies in platinum. *Journal of Physics F: Metal Physics* **1982,** *12* (7).

188. Boer, F. R. d.; Mattens, W. C. M.; Boom, R.; Miedema, A. R.; Niessen, A. K., *Cohesion in metals*. North-Holland: Netherlands, 1988.

189. Voter, A. F.; Chen, S. P., Accurate Interatomic Potentials for Ni, Al and Ni3Al. *MRS Proceedings* **1986,** *82*, 175.

190. Zhang, Y.; Ashcraft, R.; Mendelev, M. I.; Wang, C. Z.; Kelton, K. F., Experimental and molecular dynamics simulation study of structure of liquid and amorphous Ni62Nb38 alloy. *The Journal of Chemical Physics* **2016,** *145* (20), 204505.

191. Jena, P.; Sun, Q., Super Atomic Clusters: Design Rules and Potential for Building Blocks of Materials. *Chemical Reviews* **2018,** *118* (11), 5755-5870.

192. Wilcoxon, J. P.; Abrams, B. L., Synthesis, structure and properties of metal nanoclusters. *Chemical Society Reviews* **2006,** *35* (11), 1162-1194.

193. Jin, R., Atomically precise metal nanoclusters: stable sizes and optical properties. *Nanoscale* **2015,** *7* (5), 1549-1565.

194. Cramer, C. J.; Truhlar, D. G., Density functional theory for transition metals and transition metal chemistry. *Physical Chemistry Chemical Physics* **2009,** *11* (46), 10757-10816.

195. Li, G.; Jin, R., Atomically Precise Gold Nanoclusters as New Model Catalysts. *Accounts of Chemical Research* **2013,** *46* (8), 1749-1758.

196. Jia, X.; Li, J.; Wang, E., Cu Nanoclusters with Aggregation Induced Emission Enhancement. *Small* **2013,** *9* (22), 3873-3879.

197. Zhang, Y.; Song, P.; Chen, T.; Liu, X.; Chen, T.; Wu, Z.; Wang, Y.; Xie, J.; Xu, W., Unique size-dependent nanocatalysis revealed at the single atomically precise gold cluster level. *Proceedings of the National Academy of Sciences* **2018,** *115* (42), 10588.

198. Chakraborty, I.; Pradeep, T., Atomically Precise Clusters of Noble Metals: Emerging Link between Atoms and Nanoparticles. *Chemical Reviews* **2017,** *117* (12), 8208-8271.

199. Watanabe, Y., Atomically precise cluster catalysis towards quantum controlled catalysts. *Science and Technology of Advanced Materials* **2014,** *15* (6), 063501.

200. Zhu, Y.; Qian, H.; Jin, R., Catalysis opportunities of atomically precise gold nanoclusters. *Journal of Materials Chemistry* **2011,** *21* (19), 6793-6799.

201. Li, Z. Y.; Young, N. P.; Di Vece, M.; Palomba, S.; Palmer, R. E.; Bleloch, A. L.; Curley, B. C.; Johnston, R. L.; Jiang, J.; Yuan, J., Three-dimensional atomic-scale structure of size-selected gold nanoclusters. *Nature* **2008,** *451* (7174), 46-48.

202. Castleman, A. W.; Khanna, S. N., Clusters, Superatoms, and Building Blocks of New Materials. *The Journal of Physical Chemistry C* **2009,** *113* (7), 2664-2675.

203. Bartók, A. P.; De, S.; Poelking, C.; Bernstein, N.; Kermode, J. R.; Csányi, G.; Ceriotti, M., Machine learning unifies the modeling of materials and molecules. *Science Advances* **2017,** *3* (12), e1701816.

204. Jennings, P. C.; Lysgaard, S.; Hummelshøj, J. S.; Vegge, T.; Bligaard, T., Genetic algorithms for computational materials discovery accelerated by machine learning. *npj Computational Materials* **2019,** *5* (1), 46.

205. Curtis, F.; Li, X.; Rose, T.; Vázquez-Mayagoitia, Á.; Bhattacharya, S.; Ghiringhelli, L. M.; Marom, N., GAtor: A First-Principles Genetic Algorithm for

Molecular Crystal Structure Prediction. *Journal of Chemical Theory and Computation* **2018,** *14* (4), 2246-2264.

206. Wu, S. Q.; Ji, M.; Wang, C. Z.; Nguyen, M. C.; Zhao, X.; Umemoto, K.; Wentzcovitch, R. M.; Ho, K. M., An adaptive genetic algorithm for crystal structure prediction. *Journal of Physics: Condensed Matter* **2013,** *26* (3), 035402.

207. Lv, J.; Wang, Y.; Zhu, L.; Ma, Y., Particle-swarm structure prediction on clusters. *The Journal of Chemical Physics* **2012,** *137* (8), 084104.

208. Yamashita, T.; Sato, N.; Kino, H.; Miyake, T.; Tsuda, K.; Oguchi, T., Crystal structure prediction accelerated by Bayesian optimization. *Physical Review Materials* **2018,** *2* (1), 013803.

209. Yang, S.; Day, G. M., Exploration and Optimization in Crystal Structure Prediction: Combining Basin Hopping with Quasi-Random Sampling. *Journal of Chemical Theory and Computation* **2021,** *17* (3), 1988-1999.

210. Stillinger, F. H., Exponential multiplicity of inherent structures. *Physical Review E* **1999,** *59* (1), 48-51.

211. Heard, C. J.; Johnston, R. L., Global Optimisation Strategies for Nanoalloys. In *Challenges and Advances in Computational Chemistry and Physics*, Nguyen, M.; Kiran, B., Eds. Springer, Cham.: 2017; Vol. 23.

212. Davis, J. B. A.; Shayeghi, A.; Horswell, S. L.; Johnston, R. L., The Birmingham parallel genetic algorithm and its application to the direct DFT global optimisation of IrN (N = 10–20) clusters. *Nanoscale* **2015,** *7* (33), 14032-14038.

213. Garzón, I. L.; Michaelian, K.; Beltrán, M. R.; Posada-Amarillas, A.; Ordejón, P.; Artacho, E.; Sánchez-Portal, D.; Soler, J. M., Lowest Energy Structures of Gold Nanoclusters. *Physical Review Letters* **1998,** *81* (8), 1600-1603.

214. Michaelian, K.; Rendón, N.; Garzón, I. L., Structure and energetics of Ni, Ag, and Au nanoclusters. *Physical Review B* **1999,** *60* (3), 2000-2010.

215. Wang, J.; Wang, G.; Zhao, J., Density-functional study of ${\mathrm{Au}}_{n}(n=2--20)$ clusters: Lowest-energy structures and electronic properties. *Physical Review B* **2002,** *66* (3), 035418.

216. Darby, S.; Mortimer-Jones, T. V.; Johnston, R. L.; Roberts, C., Theoretical study of Cu–Au nanoalloy clusters using a genetic algorithm. *The Journal of Chemical Physics* **2002,** *116* (4), 1536-1550.

217. Sun, S.; Murray, C. B.; Weller, D.; Folks, L.; Moser, A., Monodisperse FePt Nanoparticles and Ferromagnetic FePt Nanocrystal Superlattices. *Science* **2000,** *287* (5460), 1989.

218. Kolsbjerg, E. L.; Peterson, A. A.; Hammer, B., Neural-network-enhanced evolutionary algorithm applied to supported metal nanoparticles. *Physical Review B* **2018,** *97* (19), 195424.

219. Paleico, M. L.; Behler, J., Global optimization of copper clusters at the ZnO($10\bar{1}0$) surface using a DFT-based neural network potential and genetic algorithms. *The Journal of Chemical Physics* **2020,** *153* (5), 054704.

220. Vilhelmsen, L. B.; Hammer, B., Identification of the Catalytic Site at the Interface Perimeter of Au Clusters on Rutile TiO2(110). *ACS Catalysis* **2014,** *4* (6), 1626-1631.

221. Vilhelmsen, L. B.; Walton, K. S.; Sholl, D. S., Structure and Mobility of Metal Clusters in MOFs: Au, Pd, and AuPd Clusters in MOF-74. *Journal of the American Chemical Society* **2012,** *134* (30), 12807-12816.

222. Glass, C. W.; Oganov, A. R.; Hansen, N., USPEX—Evolutionary crystal structure prediction. *Computer Physics Communications* **2006,** *175* (11), 713-720.

223. Oganov, A. R.; Glass, C. W., Crystal structure prediction using ab initio evolutionary techniques: Principles and applications. *The Journal of Chemical Physics* **2006,** *124* (24), 244704.

224. Johnston, R. L., Evolving better nanoparticles: Genetic algorithms for optimising cluster geometries. *Dalton Transactions* **2003,** (22), 4193-4207.

225. Shayeghi, A.; Götz, D.; Davis, J. B. A.; Schäfer, R.; Johnston, R. L., Pool-BCGA: a parallelised generation-free genetic algorithm for the ab initio global optimisation of nanoalloy clusters. *Physical Chemistry Chemical Physics* **2015,** *17* (3), 2104-2112.

226. Vilhelmsen, L. B.; Hammer, B., A genetic algorithm for first principles global structure optimization of supported nano structures. *The Journal of Chemical Physics* **2014,** *141* (4), 044711.

227. Deaven, D. M.; Ho, K. M., Molecular Geometry Optimization with a Genetic Algorithm. *Physical Review Letters* **1995,** *75* (2), 288-291.

228. Li, X.-T.; Yang, X.-B.; Zhao, Y.-J., Geometrical eigen-subspace framework based molecular conformation representation for efficient structure recognition and comparison. *The Journal of Chemical Physics* **2017,** *146* (15), 154108.

229. Angelo, J. E.; Moody, N. R.; Baskes, M. I., Trapping of hydrogen to lattice defects in nickel. *Modelling and Simulation in Materials Science and Engineering* **1995,** *3* (3), 289-307.

230. Williams, P. L.; Mishin, Y.; Hamilton, J. C., An embedded-atom potential for the Cu–Ag system. *Modelling and Simulation in Materials Science and Engineering* **2006,** *14* (5), 817-833.

231. Mishin, Y., Atomistic modeling of the γ and γ′-phases of the Ni–Al system. *Acta Materialia* **2004,** *52* (6), 1451-1467.

232. Olsson, P. A. T., Transverse resonant properties of strained gold nanowires. *Journal of Applied Physics* **2010,** *108* (3), 034318.

233. Mishin, Y.; Farkas, D.; Mehl, M. J.; Papaconstantopoulos, D. A., Interatomic potentials for monoatomic metals from experimental data and ab initio calculations. *Physical Review B* **1999,** *59* (5), 3393-3407.

234. Mendelev, M. I.; Kramer, M. J.; Hao, S. G.; Ho, K. M.; Wang, C. Z., Development of interatomic potentials appropriate for simulation of liquid and glass properties of NiZr2 alloy. *Philosophical Magazine* **2012,** *92* (35), 4454-4469.

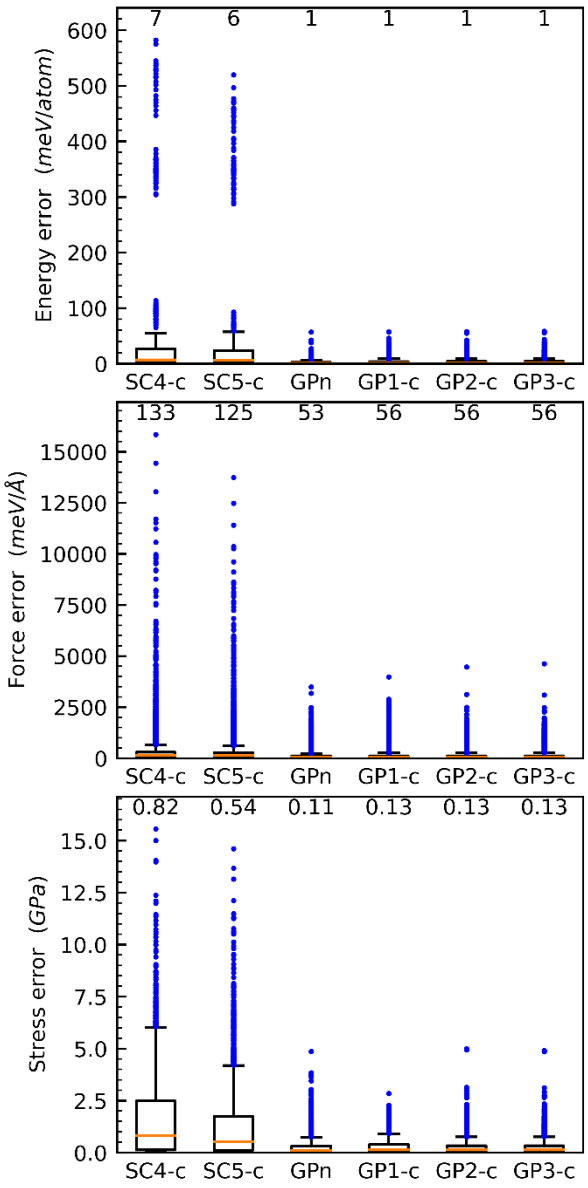
Vita

Alberto Hernandez was born in December 1991, in Cartago, Costa Rica. In 2015, he received his B.Eng. degree in Materials Science and Engineering from the Costa Rica Institute of Technology. In the same year, he joined the PhD program in Materials Science and Engineering at The Johns Hopkins University, under the guidance of Prof. Tim Mueller. In the Mueller Research Group, he developed POET, a machine learning approach to develop simple, fast, and accurate interatomic potential models using symbolic regression, and with the Mueller Group he demonstrated how the models developed with POET for copper transfer well to other elemental systems that are chemically similar to copper. With the Mueller Research Group, he helped develop a database of atomically precise nanoclusters, named the Quantum Cluster Database, which is the largest database of these type of materials at the density functional theory level of accuracy, and includes thousands of novel clusters.

Appendix A

Table-Appx. 1. POET GPn models with full numerical precision

Element	Seed	POET GPn model
Cu	GP1	$64.15515940524648\Sigma\left(\left(\frac{1}{r^{4.461239524986285}}-0.1954983463474226^{r}\right)f(r)\right)+\frac{29.76431804908527}{\Sigma(f(r))}$
Ag	SC	$\Sigma\left(\left(\frac{424.9965825900254}{r^{7.322698914792303}}-0.01102677991908652\right)f(r)\right)+8.172810046531140\times0.7017644070210753^{\Sigma\left(r^{1-r}f(r)\right)}$
Au	SC	$\Sigma\left(\left(\frac{8167.21232353391}{r^{10.8596729498766}}-0.01085427302578209\right)f(r)\right)+\frac{0.05350843770358775}{\Sigma\left(r^{-5.377144047422479}f(r)\right)}$
Ni	SC	$\Sigma\left(\left(\frac{43.844638787564}{r^{3.69765898135618}}-0.1481588631749128\right)f_2(r)\right)-62.55959167587809\left(\Sigma\left(re^{-1.885228718370605r}f_2(r)\right)\right)^{0.870093496934648}$
Pd	SC	$42.61343703107455\Sigma\left(r^{2.135723519763371r}f(r)\right)-42.61343703107455\left(\Sigma\left(r^{-4.790271382259407}f(r)\right)\right)^{0.070787893131414}$
Pt	GP2	$\Sigma\left(\left(10.89243794141817r^{5.036647157219199}-3.649317739301988^{r}-0.0373360678671021\right)f_2(r)\right)+12.720343347841920\times0.2146539628086967^{\Sigma\left(3.649317739301988r^{-r}f_2(r)\right)}$
Rh	No seed	$\Sigma\left(\left(-89.55102711997160\times0.2640482761663984^{r}-0.3169171347295917+\frac{98.9083181938882}{r^{3.580044561312461}}\right)f(r)\right)+\frac{0.0826182489861164}{\Sigma\left(0.1359924613515034^{r}f(r)\right)}$
Ir	SC	$\Sigma\left(28.22630935153837r^{-1.941760188276659^{r}}f(r)\right)+\frac{78.24327611794345}{\Sigma(f(r))}$



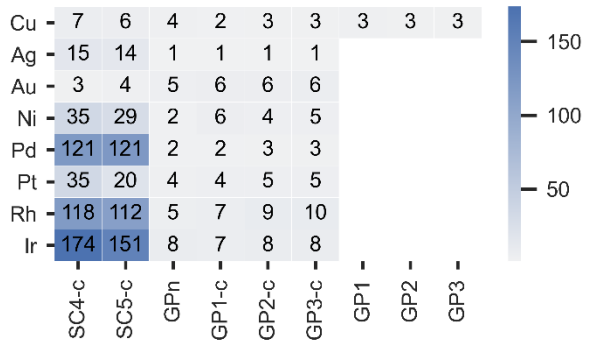
SI-Figure 1. Distribution of errors for models across Cu, Ag, Au, Ni, Pd, Pt, Rh, and Ir

SI-Table 1. References of literature models

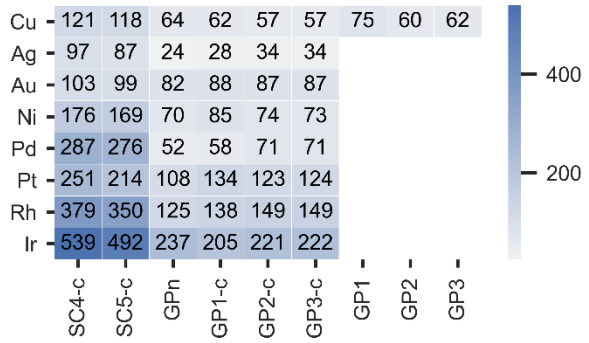
Model	DOI	Ref. number
Asta_Cu_ABCHM	https://doi.org/10.1080/14786430802206482	140
Asta_Cu_Cu1	https://doi.org/10.1080/14786430802206482	140
Baskes_Ni	https://doi.org/10.1088/0965-0393/3/3/001	229
Chen_Ag_SC	https://doi.org/10.1080/09500839008206493	138

Chen_Au_SC	https://doi.org/10.1080/09500839008206493	138
Chen_Cu_SC	https://doi.org/10.1080/09500839008206493	138
Chen_Ir_SC	https://doi.org/10.1080/09500839008206493	138
Chen_Ni	https://doi.org/10.1557/PROC-82-175	189
Chen_Ni_SC	https://doi.org/10.1080/09500839008206493	138
Chen_Pd_SC	https://doi.org/10.1080/09500839008206493	138
Chen_Pt_SC	https://doi.org/10.1080/09500839008206493	138
Chen_Rh_SC	https://doi.org/10.1080/09500839008206493	138
Daw_Ag	https://doi.org/10.1103/physrevb.33.7983	144
Daw_Au	https://doi.org/10.1103/physrevb.33.7983	144
Daw_Cu	https://doi.org/10.1103/physrevb.33.7983	144
Daw_Ni	https://doi.org/10.1103/physrevb.33.7983	144
Daw_Pt	https://doi.org/10.1103/physrevb.33.7983	144
Durukanoglu_Cu	https://doi.org/10.1088/0953-8984/26/3/035404	136
Durukanoglu_Ni	https://doi.org/10.1088/0953-8984/26/3/035404	136
Finnis_Ag	https://doi.org/10.1080/01418618708204485	175
Finnis_Au	https://doi.org/10.1080/01418618708204485	175
Finnis_Cu	https://doi.org/10.1080/01418618708204485	175
Finnis_Ni	https://doi.org/10.1080/01418618708204485	175
Foiles_Au	https://doi.org/10.1007/s10853-017-1706-1	172
Foiles_Pt	https://doi.org/10.1007/s10853-017-1706-1	172
Hamilton_Ag	https://doi.org/10.1088/0965-0393/14/5/002	230
Kelton_Ni	http://dx.doi.org/10.1063/1.4968212	190
Kress_Cu_EAM1	https://doi.org/10.1103/physrevb.63.224106	123
Kress_Cu_EAM2	https://doi.org/10.1103/physrevb.63.224106	123
Mishin_Ni	https://doi.org/10.1016/j.actamat.2003.11.026	231
Olsson_Au	https://doi.org/10.1063/1.3460127	232
Papaconstantopoulos_Ni	https://doi.org/10.1103/physrevb.59.3393	233
Sheng_Ag	https://doi.org/10.1103/PhysRevB.83.134118	176
Sheng_Au	https://doi.org/10.1103/PhysRevB.83.134118	176
Sheng_Cu	https://doi.org/10.1103/PhysRevB.83.134118	176
Sheng_Ir	https://doi.org/10.1103/PhysRevB.83.134118	176
Sheng_Ni	https://doi.org/10.1103/PhysRevB.83.134118	176
Sheng_Pd	https://doi.org/10.1103/PhysRevB.83.134118	176
Sheng_Pt	https://doi.org/10.1103/PhysRevB.83.134118	176
Sheng_Rh	https://doi.org/10.1103/PhysRevB.83.134118	176
Wang_Ni	https://doi.org/10.1080/14786435.2012.712220	234
Wolfer_Ag	https://doi.org/10.1557/JMR.1989.0102	143
Wolfer_Au	https://doi.org/10.1557/JMR.1989.0102	143
Wolfer_Cu	https://doi.org/10.1557/JMR.1989.0102	143
Wolfer_Ni	https://doi.org/10.1557/JMR.1989.0102	143

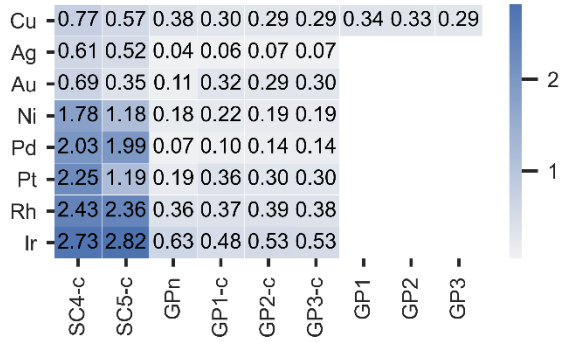
Wolfer_Pt	https://doi.org/10.1557/JMR.1989.0102	143



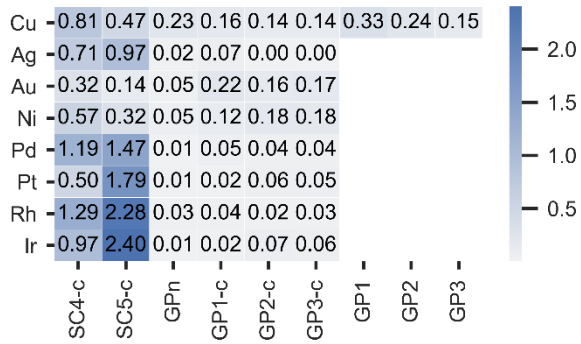
SI-Figure 2. Energy MAE on validation in meV/atom



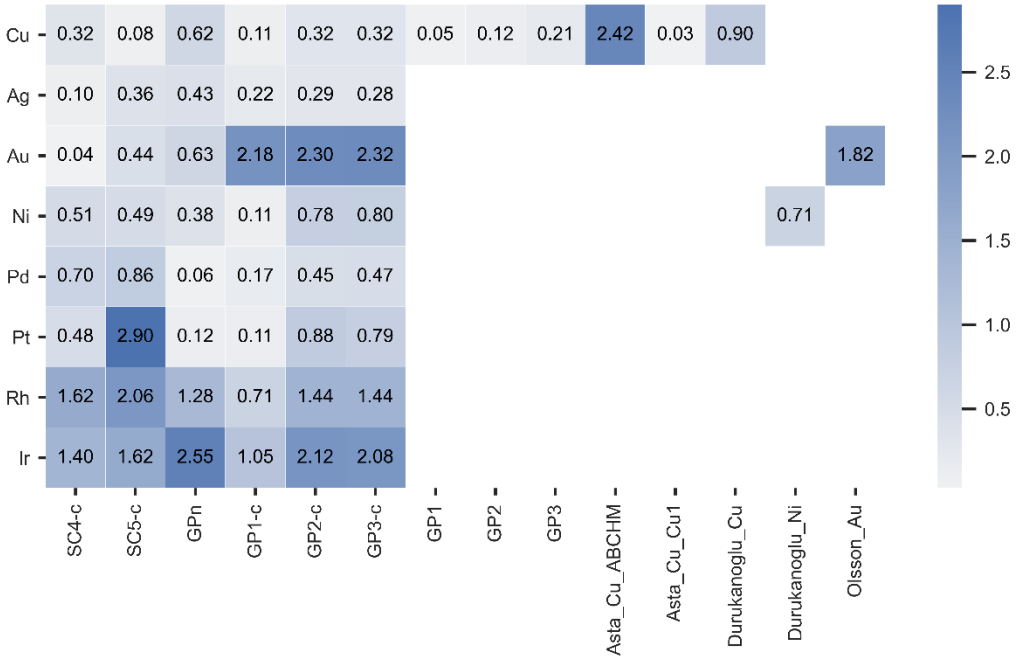
SI-Figure 3. Force MAE on validation in meV/Å



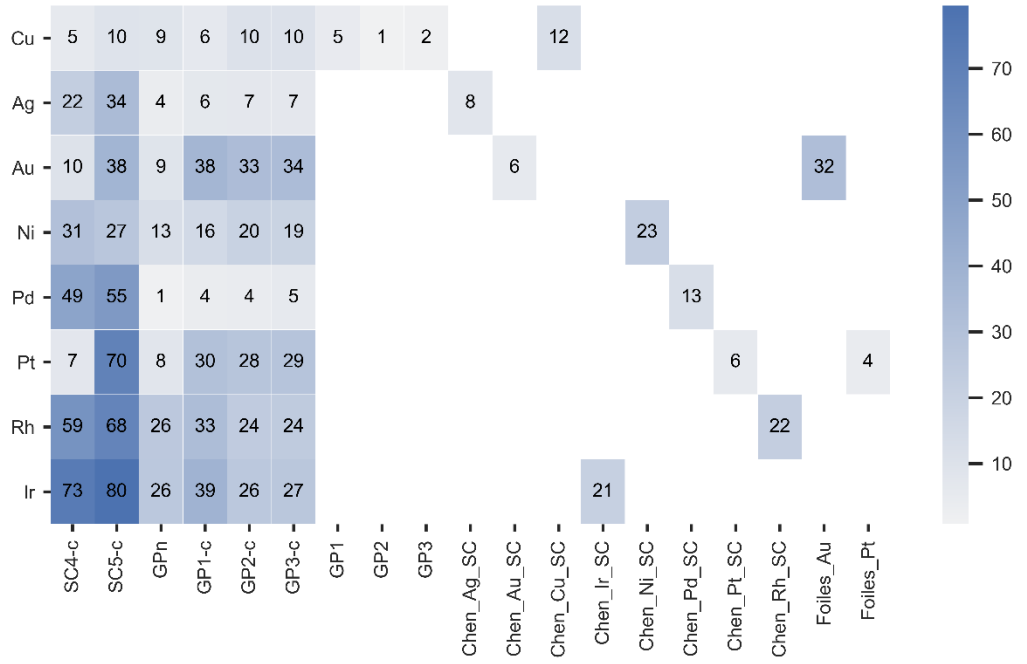
SI-Figure 4. Stress MAE in GPa



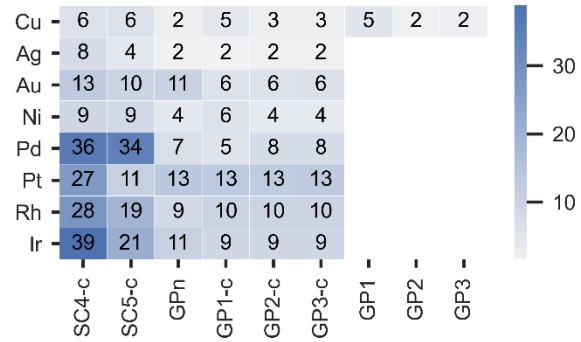
SI-Figure 5. Absolute percent error on fcc lattice parameter



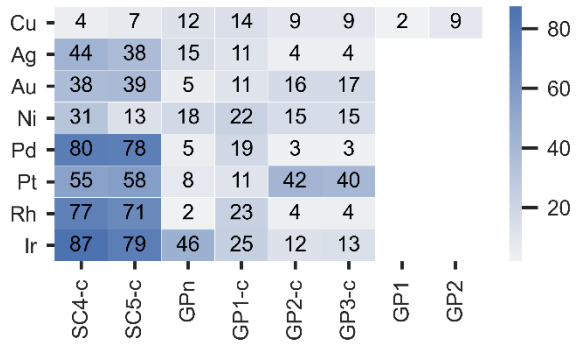
SI-Figure 6. Absolute percent error on bcc lattice parameter.



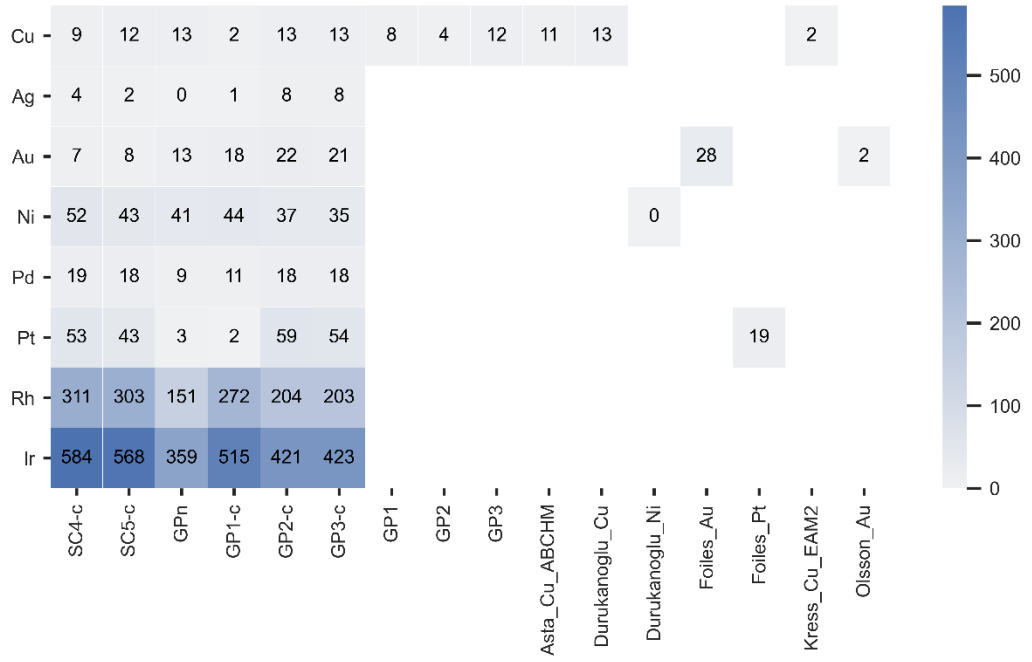
SI-Figure 7. Mean absolute percent error on the fcc elastic constants C11, C12 and C44.



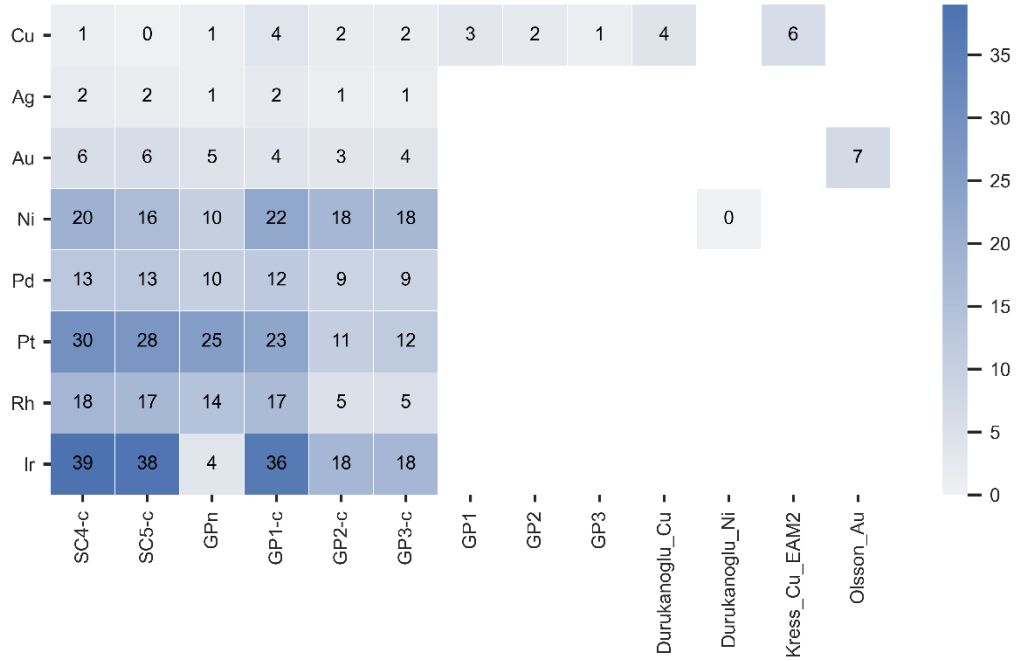
SI-Figure 8. Mean absolute percent error on phonon frequencies.



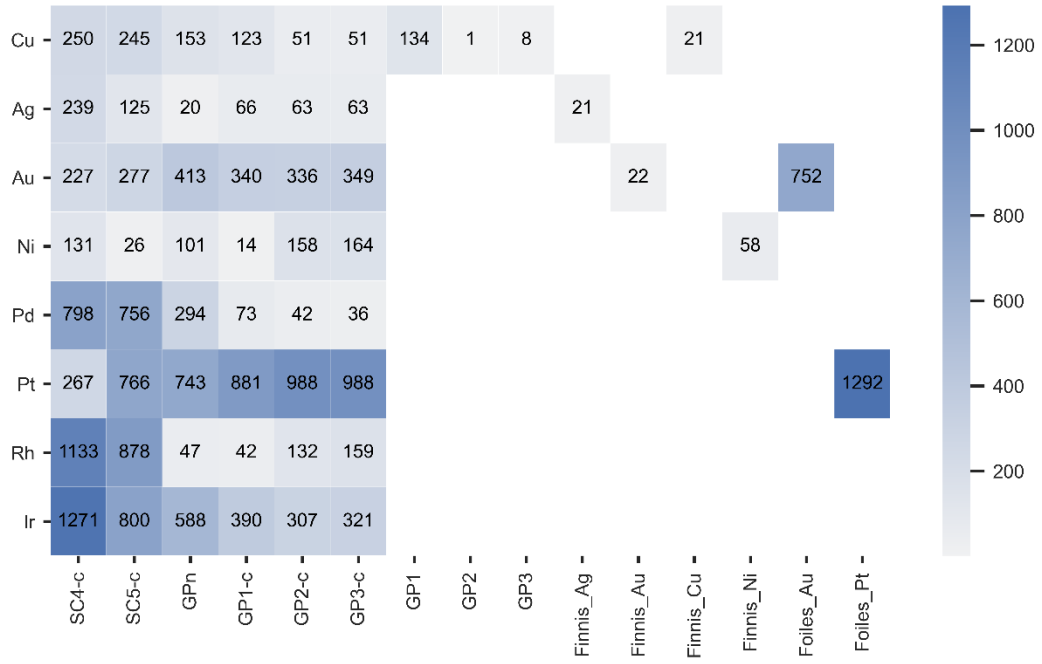
SI-Figure 9. Mean absolute percent error on 13-low index surface energies.



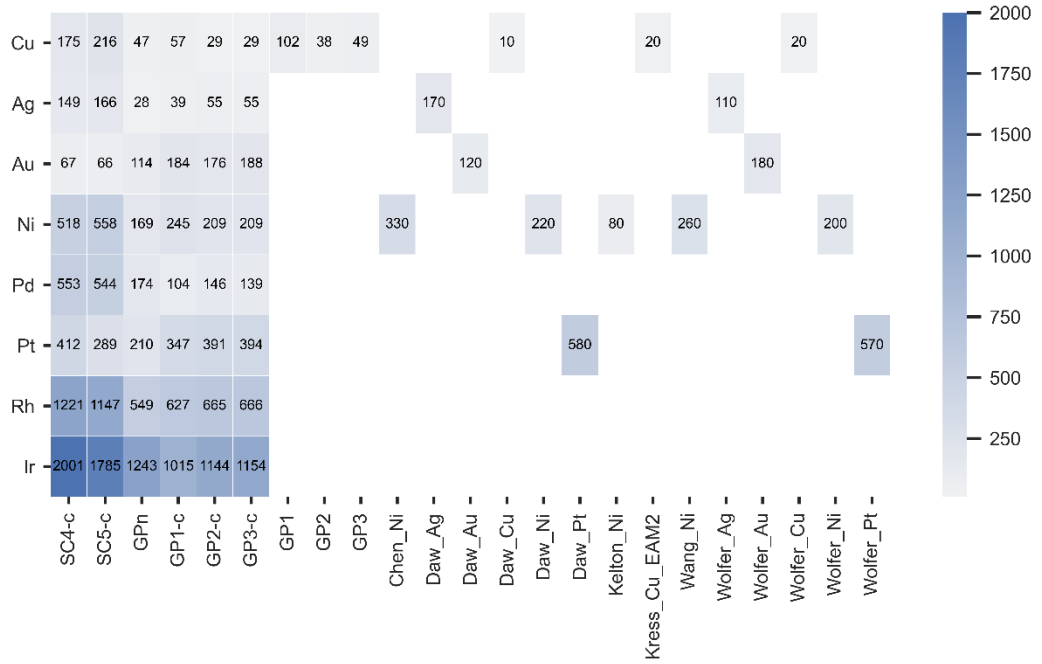
SI-Figure 10. Absolute error on bcc formation energy in meV



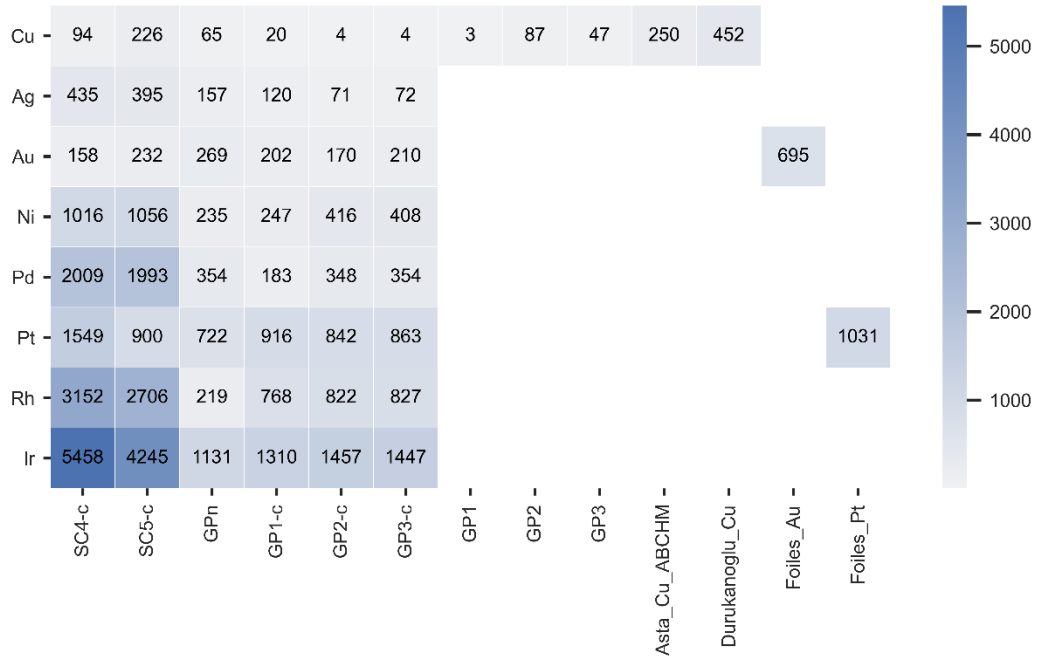
SI-Figure 11. Absolute error on hcp formation energy in meV



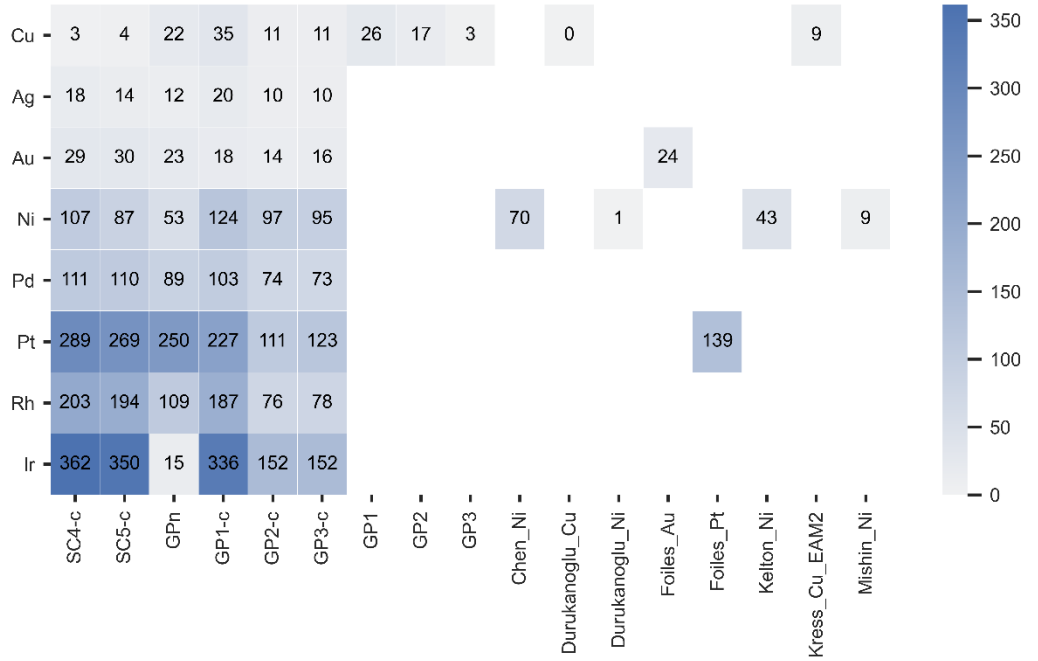
SI-Figure 12. Absolute error on the vacancy formation energy in meV



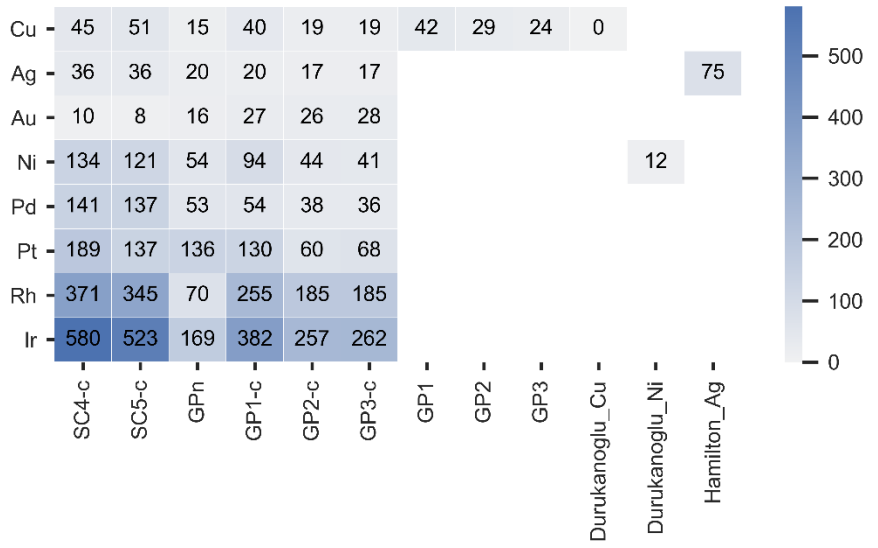
SI-Figure 13. Absolute error on the vacancy migration energy in meV



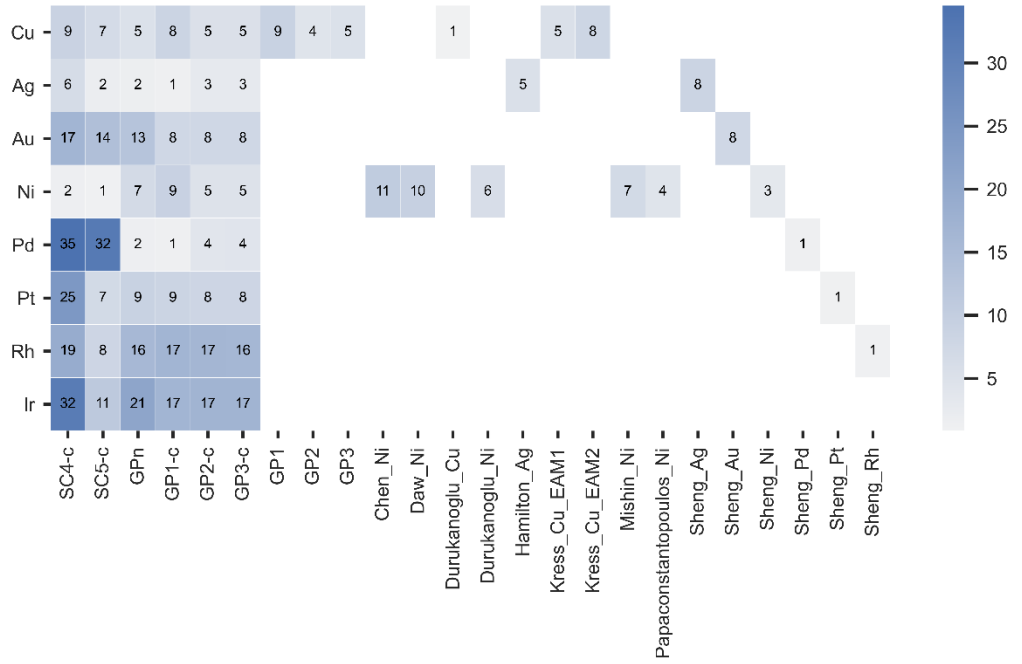
SI-Figure 14. Absolute error on the <100> dumbbell formation energy in meV



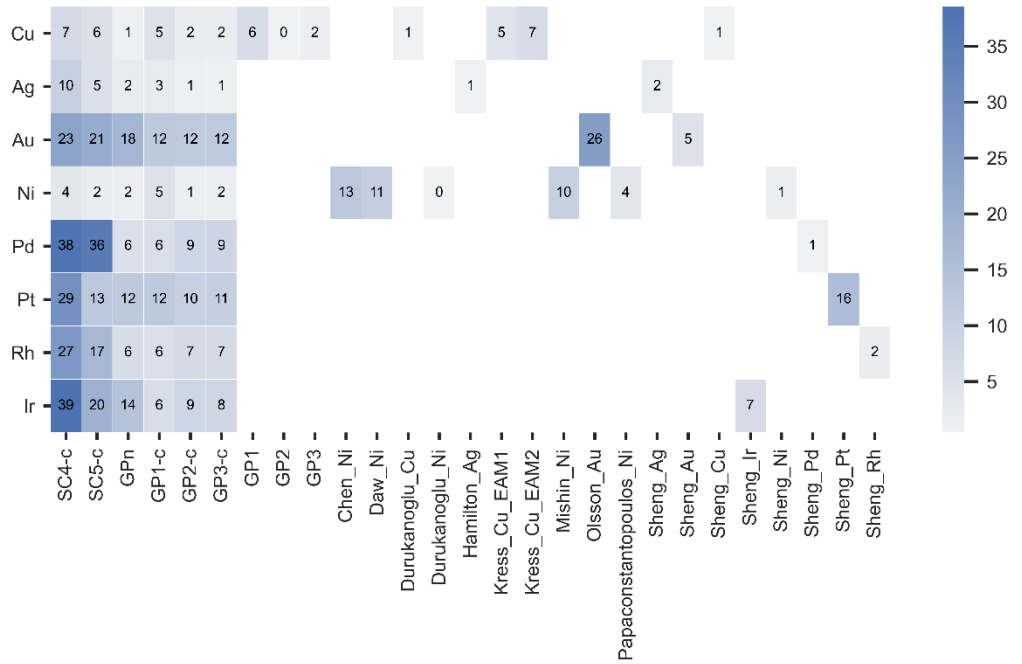
SI-Figure 15. Absolute error on the intrinsic stacking fault energy in mJ/m^2



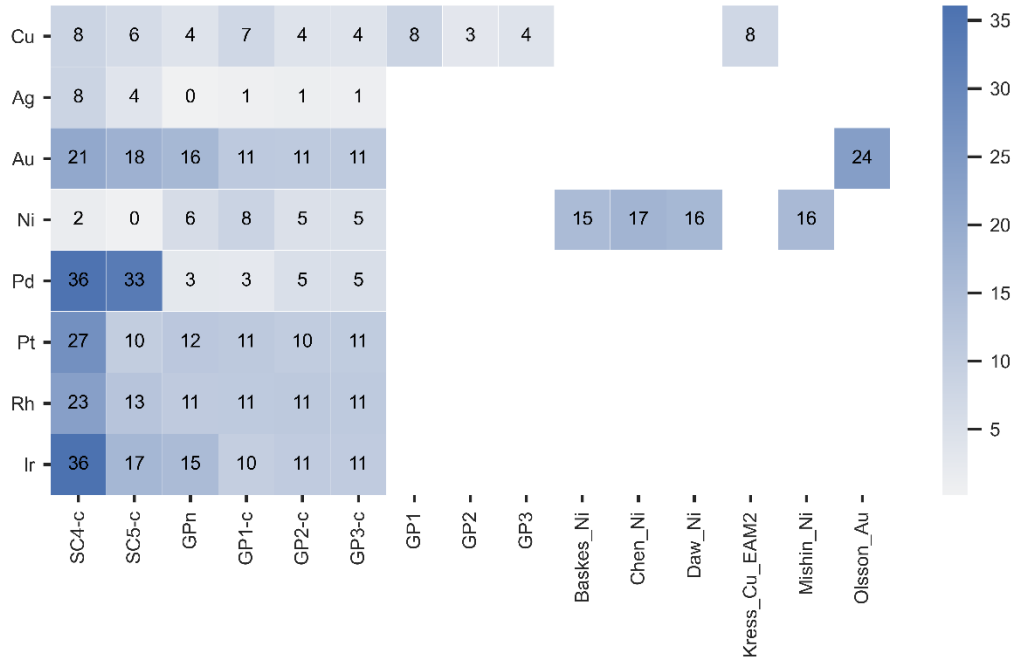
SI-Figure 16. Absolute error on the unstable stacking fault energy in mJ/m^2



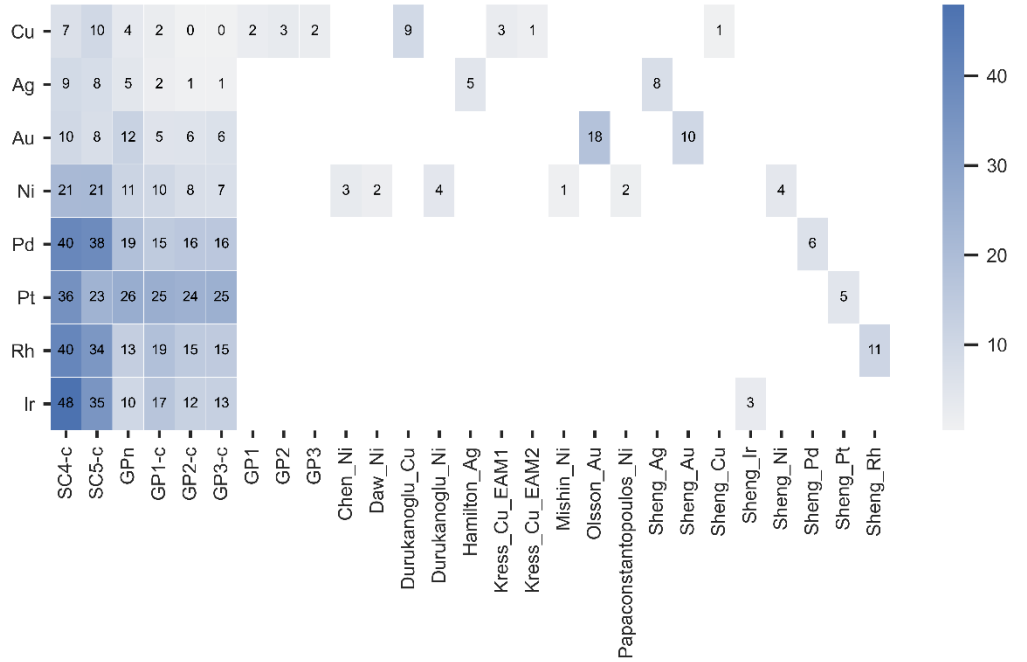
SI-Figure 17. Absolute percent error on the $\nu_L(K)$ phonon frequency.



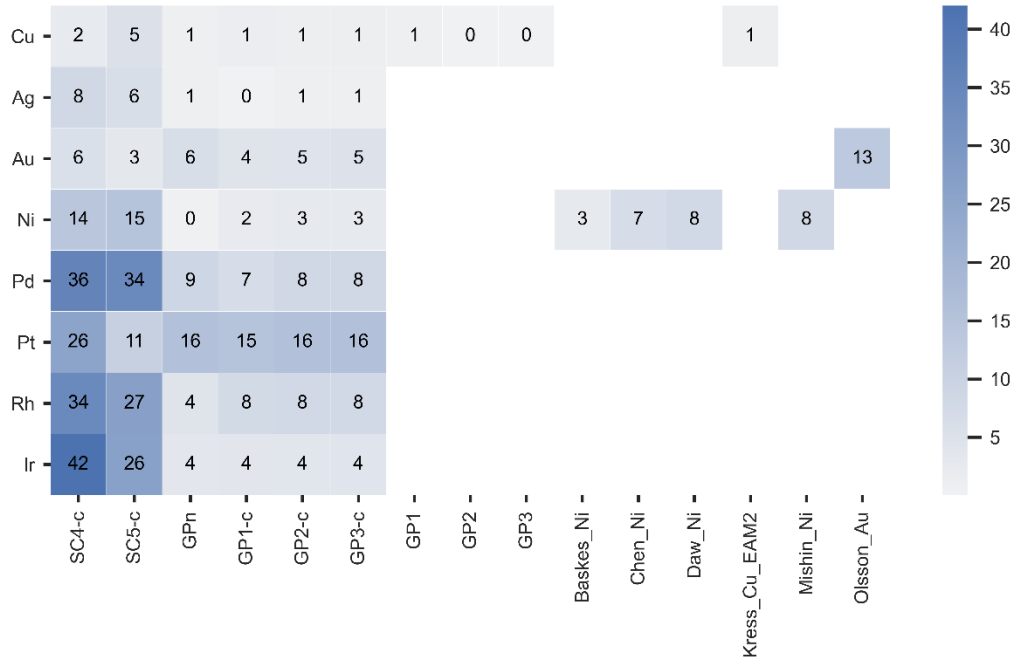
SI-Figure 18. Absolute percent error on the $\nu_L(L)$ phonon frequency.



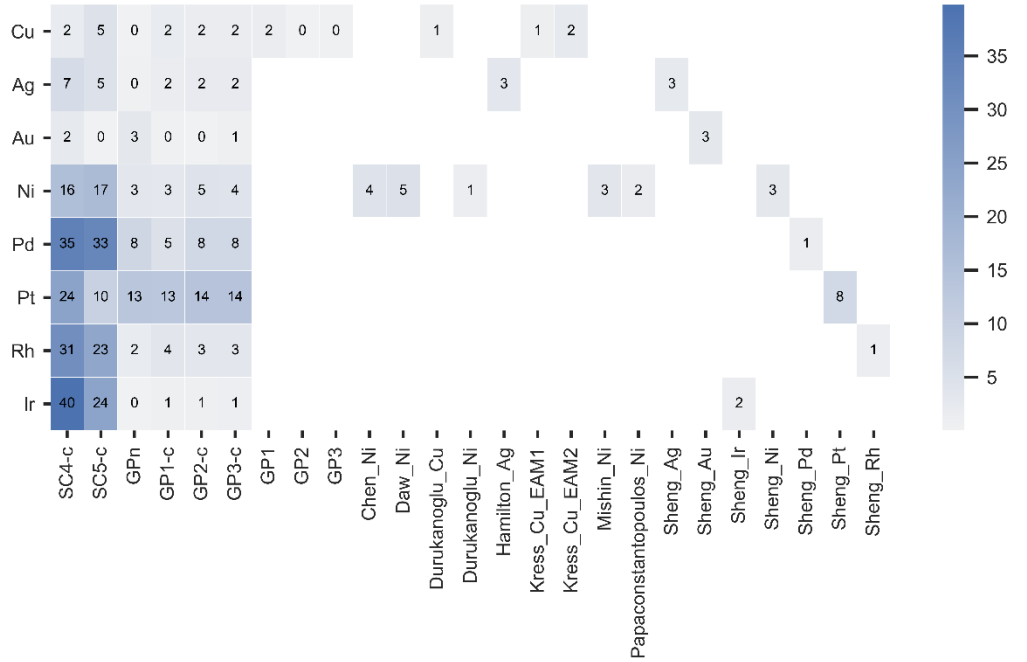
SI-Figure 19. Absolute percent error on the $\nu_L(X)$ phonon frequency.



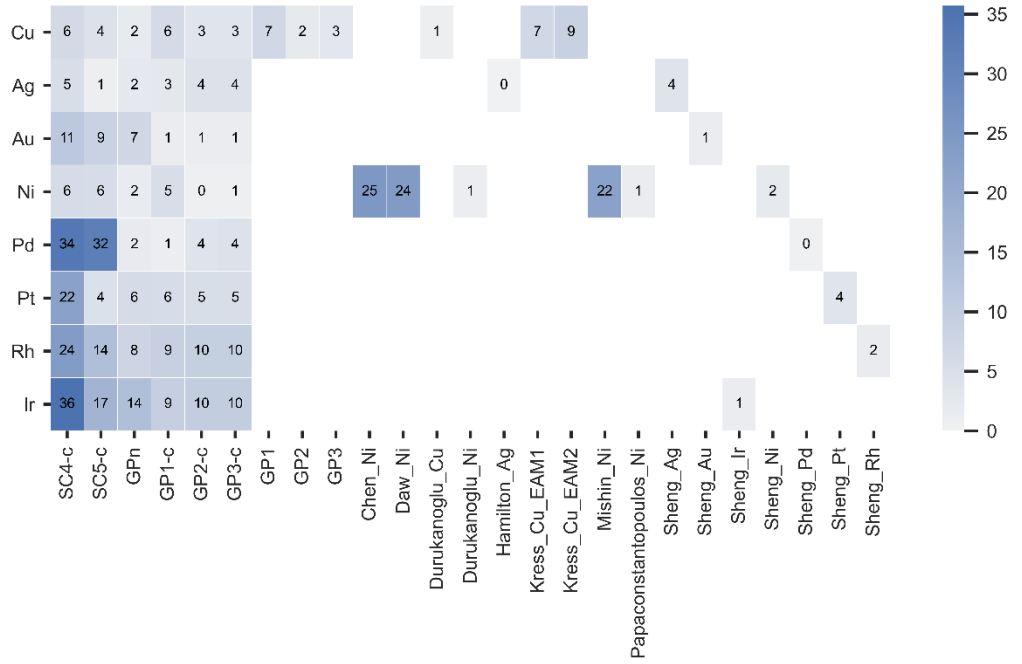
SI-Figure 20. Absolute percent error on the $\nu_T(L)$ phonon frequency.



SI-Figure 21. Absolute percent error on the $\nu_T(X)$ phonon frequency.



SI-Figure 22. Absolute percent error on the $\nu_{T1}(K)$ phonon frequency.



SI-Figure 23. Absolute percent error on the ν_{T2}(K) phonon frequency.

Table-Appx. 2. Parameters of SC models

element	x0	x1	x2	x3	x4
Cu	639.4431951	-9.314032349	8.539842097	-3.939504085	0.5
Ag	27844.05026	-12.47480779	12.68528807	-5.625307263	0.5
Au	8181.184056	-9.934009998	109.9086417	-7.831419368	0.5
Ni	659.8848777	-9.650559919	9.100240433	-4.223886938	0.5
Pd	25084.21852	-12.89563225	38.1980012	-8.354901196	0.5
Pt	8491.995124	-10.06919047	150.4289378	-8.108452359	0.5
Rh	22382.02447	-13.0556093	23.92776153	-7.120089457	0.5
Ir	185595.7209	-15.40871138	32.63989534	-8.445864459	0.5

The parameters of SC correspond to: $E_i = \sum_j x_0 r^{x1} f(r) - \left(\sum_j x_2 r^{x_3} f(r) \right)^{x_4}$

Table-Appx. 3. Parameters of SCa models

element	x0	x1	x2	x3	x4
Cu	631.267176	-9.406630905	38.29001209	-3.519277139	0.064366175
Ag	27857.11891	-12.64781229	7.159344695	-4.176935801	0.942883625
Au	8173.588449	-10.2864238	111.0000883	-7.058621328	0.705163433
Ni	641.8157332	-9.768708674	42.24618404	-3.296206018	0.059023287

Pd	25086.89713	-13.11906214	38.52709442	-7.439547099	0.670360942
Pt	8499.317134	-11.02098225	141.8100625	-5.068173828	1.762037756
Rh	22381.24106	-13.32746526	23.52746732	-5.130588742	1.240077883
Ir	185597.0729	-15.64936747	29.98698996	-4.676368063	1.887777152

The parameters of SC correspond to: $E_i = \sum_j x_0 r^{x1} f(r) - \left(\sum_j x_2 r^{x_3} f(r) \right)^{x_4}$

Appendix B

Table-Appx. 4. References of the clusters obtained from the literature for the Quantum Cluster Database

El.	DOIs
Ag	https://doi.org/10.1039/A706221D; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9; https://doi.org/10.1021/ACS.JPCC.6B10404; https://doi.org/10.1103/PhysRevB.70.165403; https://doi.org/10.1021/JP404493W; https://doi.org/10.1039/A709249K
Al	https://doi.org/10.1039/A706221D; http://dx.doi.org/10.1021/acs.jpca.9b09309; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1063/1.3075834; https://doi.org/10.1007/978-94-009-0211-4_9; https://doi.org/10.1021/ACS.JPCC.6B10404; https://doi.org/10.1063/1.1574797
Au	https://doi.org/10.1002/jcc.21980; https://doi.org/10.1103/PhysRevB.72.205428; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1103/PhysRevB.70.165403; https://doi.org/10.1016/j.physleta.2009.12.032; https://doi.org/10.1039/A709249K
B	https://doi.org/10.1039/C5CP01851J; https://doi.org/10.1039/C3CC48392D; https://doi.org/10.1016/j.cplett.2014.05.069; https://doi.org/10.1039/C5CC09111J; https://doi.org/10.1016/j.cplett.2013.05.041; https://pubs.acs.org/doi/10.1021/jp9085848; https://doi.org/10.1039/C4CP02323D
Ba	https://doi.org/10.1039/A706221D; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Be	https://doi.org/10.1155/2012/648386
Ca	https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1039/A706221D; https://doi.org/10.1007/978-94-009-0211-4_9; https://doi.org/10.1155/2012/648386; https://doi.org/10.1063/1.470729
Cd	https://doi.org/10.1016/j.commatsci.2004.07.009; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1140/epjd/e2007-00092-x; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1103/PhysRevB.68.195418; https://doi.org/10.1039/B406562J; https://doi.org/10.1039/C6CP04948F
Co	https://doi.org/10.1063/1.1940028; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Cr	https://doi.org/10.1039/A706221D; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1063/1.470729;

	https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Cs	https://doi.org/10.1039/A706221D; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Cu	https://doi.org/10.1039/A706221D; https://doi.org/10.1016/j.comptc.2013.06.014; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9; https://doi.org/10.1103/PhysRevB.70.165403; https://doi.org/10.1039/A709249K
Fe	https://doi.org/10.1039/A706221D; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1080/14786430903270668; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Ga	https://doi.org/10.1039/C2NR31222K; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1063/1.3615501
Ge	https://doi.org/10.1063/1.2192783
Hf	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Hg	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
In	https://doi.org/10.1021/acs.jpcc.6b10404
Ir	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
K	https://doi.org/10.1039/A706221D; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Li	https://doi.org/10.1016/j.cplett.2015.11.049
Mg	https://doi.org/10.1016/j.comptc.2016.01.011; https://doi.org/10.1155/2012/648386; https://doi.org/10.1021/acs.jpcc.6b10404; http://dx.doi.org/10.1021/acs.jpca.9b09309
Mn	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Mo	https://doi.org/10.1039/A706221D; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1080/14786430903270668; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Na	https://doi.org/10.1039/A706221D; https://doi.org/10.1021/acs.jpcc.6b10404; http://dx.doi.org/10.1021/acs.jpca.9b09309; https://doi.org/10.1103/PhysRevB.68.195418; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9; https://doi.org/10.1016/j.commatsci.2004.07.009; https://doi.org/10.1140/epjd/e2007-00092-x
Nb	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404

Ni	https://doi.org/10.1039/A706221D; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9; https://doi.org/10.1016/j.comptc.2011.09.028; https://doi.org/10.1039/A709249K
Os	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
P	http://dx.doi.org/10.1021/acs.jpca.9b09309
Pb	https://doi.org/10.1039/A706221D; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9; https://doi.org/10.1140/epjd/e2002-00232-x; https://doi.org/10.1039/c6nr02080a
Pd	https://doi.org/10.1039/B303347C; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Pt	https://doi.org/10.1039/A709249K; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Rb	https://doi.org/10.1039/A706221D; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Re	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Rh	https://doi.org/10.1039/A709249K; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Ru	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Sc	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Se	https://doi.org/10.1016/j.comptc.2012.02.031
Si	https://doi.org/10.1103/PhysRevA.69.053202; https://doi.org/10.1063/1.2191494; https://doi.org/10.1063/1.2165181; https://doi.org/10.1002/anie.200461753; https://doi.org/10.1103/PhysRevLett.95.055501
Sr	https://doi.org/10.1039/A706221D; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Ta	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Ti	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Tl	https://doi.org/10.1021/acs.jpcc.6b10404
V	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
W	https://doi.org/10.1039/A706221D; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1063/1.470729; https://doi.org/10.1088/0953-4075/29/21/002; https://doi.org/10.1007/978-94-009-0211-4_9
Y	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404
Zn	https://doi.org/10.1039/C8NR05517C; https://doi.org/10.1016/j.commatsci.2004.07.009; https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1140/epjd/e2007-00092-x; https://doi.org/10.1021/acs.jpcc.6b10404; https://doi.org/10.1103/PhysRevB.68.195418
Zr	https://doi.org/10.1039/C7CP02240A; https://doi.org/10.1021/acs.jpcc.6b10404

Table-Appx. 5. Pseudopotentials used in VASP.

Element	Name (TITEL from POTCAR)
Ag	PAW_PBE Ag 02Apr2005
Al	PAW_PBE Al 04Jan2001
As	PAW_PBE As 22Sep2009
Au	PAW_PBE Au 04Oct2007
B	PAW_PBE B 06Sep2000
Ba	PAW_PBE Ba_sv 06Sep2000
Be	PAW_PBE Be 06Sep2000
Bi	PAW_PBE Bi 08Apr2002
Br	PAW_PBE Br 06Sep2000
C	PAW_PBE C 08Apr2002
Ca	PAW_PBE Ca_pv 06Sep2000
Cd	PAW_PBE Cd 06Sep2000
Cl	PAW_PBE Cl 06Sep2000
Co	PAW_PBE Co 02Aug2007
Cr	PAW_PBE Cr 06Sep2000
Cs	PAW Cs_sv_GW 23Mar2010
Cu	PAW_PBE Cu 22Jun2005
F	PAW_PBE F 08Apr2002
Fe	PAW_PBE Fe 06Sep2000
Ga	PAW_PBE Ga 08Apr2002
Ge	PAW_PBE Ge 05Jan2001
Hf	PAW_PBE Hf 20Jan2003
Hg	PAW_PBE Hg 06Sep2000
I	PAW_PBE I 08Apr2002
In	PAW_PBE In 08Apr2002
Ir	PAW_PBE Ir 06Sep2000
K	PAW_PBE K_pv 17Jan2003
Li	PAW_PBE Li 17Jan2003
Mg	PAW_PBE Mg 13Apr2007
Mn	PAW_PBE Mn 06Sep2000
Mo	PAW_PBE Mo 08Apr2002
N	PAW_PBE N 08Apr2002
Na	PAW_PBE Na 08Apr2002
Nb	PAW_PBE Nb_pv 08Apr2002
Ni	PAW_PBE Ni 02Aug2007
O	PAW_PBE O 08Apr2002

Os	PAW_PBE Os 17Jan2003
P	PAW_PBE P 06Sep2000
Pb	PAW_PBE Pb 08Apr2002
Pd	PAW_PBE Pd 04Jan2005
Pt	PAW_PBE Pt 04Feb2005
Rb	PAW_PBE Rb_pv 06Sep2000
Re	PAW_PBE Re 17Jan2003
Rh	PAW_PBE Rh 04Feb2005
Ru	PAW_PBE Ru 04Feb2005
S	PAW_PBE S 06Sep2000
Sb	PAW_PBE Sb 06Sep2000
Sc	PAW_PBE Sc 04Feb2005
Se	PAW_PBE Se 06Sep2000
Si	PAW_PBE Si 05Jan2001
Sn	PAW_PBE Sn 08Apr2002
Sr	PAW_PBE Sr_sv 07Sep2000
Ta	PAW_PBE Ta 17Jan2003
Te	PAW_PBE Te 08Apr2002
Ti	PAW_PBE Ti 08Apr2002
Tl	PAW_PBE Tl 08Apr2002
V	PAW_PBE V 08Apr2002
W	PAW_PBE W 08Apr2002
Y	PAW_PBE Y_sv 25May2007
Zn	PAW_PBE Zn 06Sep2000
Zr	PAW_PBE Zr_sv 04Jan2005